

From Human Motion to Humanoid Control : Laboratory report

Nicolas JS Testard, Josep Rueda

2020

Contents

1	Introduction	3
2	Pendulum model	3
2.1	Single pendulum	3
2.1.1	Kinematic model	3
2.1.2	Dynamic model	4
2.1.3	Kinematics planner	5
2.1.4	Energy Computation	5
2.1.5	Simulation and animation	6
2.2	Double pendulum	7
2.2.1	Kinematic model	7
2.2.2	Dynamic model	8
2.2.3	Kinematics planner	10
2.2.4	Energy Computation	10
2.2.5	Simulation and animation	11
3	Biomechanical model	12
4	Data acquisition	15
4.1	Capture system	15
4.2	Data provided	16
4.2.1	Hanavan parameters	16
4.2.2	Motion and force plate data	17
5	Use of the data	18
5.1	Construction of the biomechanical model bodies	18
5.2	Use of the motion data	19

6 Computation of the velocities and the accelerations	21
6.1 Angular velocity and acceleration	21
6.2 Linear velocity and acceleration	23
6.3 Filtering	24
7 The Newton Euler algorithm	24
7.1 Newton Euler algorithm for one body	25
7.2 Use of the Newton-Euler algorithm for one body	26
7.2.1 Computation of the joint torques	26
7.2.2 Computation of the ground reaction efforts	27
8 Analysis of the results	27
8.1 Alignment of the reference frames and first analysis	27
8.2 Influence of the dynamics on the noise	30
8.3 Computation of joint torques	34
8.4 Study of the energy for the jumping motions	35
9 Conclusion	37
A Folder organization	38
B Workflow	39
B.1 Pendulum	39
B.2 Hanavan.m	40
B.3 Animation.m	41
B.4 Ne_fullbody.m	42
B.5 Ne_arm.m	43
C Ground reaction forces for some motion	44

1. Introduction

The purpose of this lab is to study how to determine the efforts implied in a human motion from a set of given kinematics data. This data of the motion of a human body come from a camera motion capture.

First, simple examples of the computation of torques with the Newton-Euler method will be detailed for the case of the single pendulum and double pendulum models. Then, it will be presented how to modelize a human body through the modified Hanavan model and how to apply this model to a specific subject. Afterwards, using the data provided (angle and position of the bodies), the techniques to recover the velocity and acceleration with few noises will be presented. Moreover, the Newton-Euler algorithm will be extended to the whole body and finally the result of this method will be presented, analyzed and compared to given ground reaction efforts data from a force plate.

The organization of the folder containing the codes is presented in the Appendix A.

2. Pendulum model

2.1 Single pendulum

A 2D inverted pendulum is considered: a point mass m attached to a rotational joint by a mass-less rod:

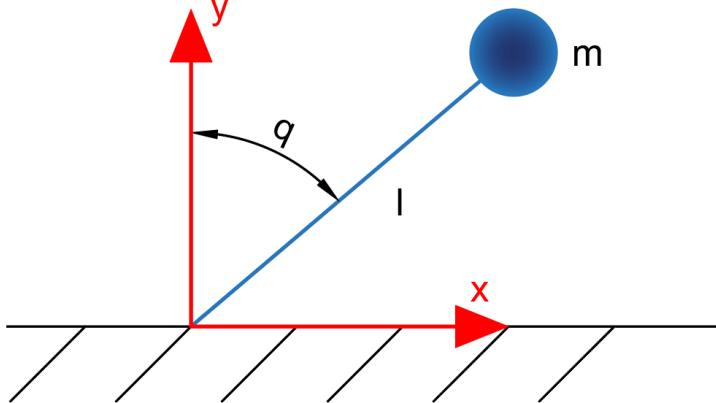


Figure 1: Single pendulum scheme

Where m is the mass of the pendulum, l is the length of the link, and q is the angle between the link and the y global axis.

2.1.1 Kinematic model

The equations of the position of the mass depending on the angle position are computed.

$$x = -l \sin(q) \quad (1)$$

$$y = l \cos(q) \quad (2)$$

Where x and y are the position of the mass point m (of mass m) with respect to the origin.

Thereafter the equations of the velocity of the mass point depending on the angle position and angular velocity are obtained by deriving the previous 2 equations.

$$\dot{x} = -l \cos(q) \dot{q} \quad (3)$$

$$\dot{y} = -l \sin(q) \dot{q} \quad (4)$$

Where \dot{x} and \dot{y} are the velocity of the mass m with respect to the origin.

The acceleration of the mass depending on the angle position, angular velocity and angular acceleration are finally obtained by deriving the velocity equations.

$$\ddot{x} = l \sin(q) \dot{q}^2 - l \cos(q) \ddot{q} \quad (5)$$

$$\ddot{y} = -l \cos(q) \dot{q}^2 - l \sin(q) \ddot{q} \quad (6)$$

Where \ddot{x} and \ddot{y} are the acceleration of the mass point m with respect to the origin.

2.1.2 Dynamic model

To solve the dynamic model, the Newton-Euler approach has been used. This method relies on applying the two following equations to each singular body:

For the Forces:

$$\sum F = m_j \cdot \dot{v}_j + \dot{\omega}_j \times ms_j + \omega_j \times (\omega_j \times ms_j) \quad (7)$$

For the Torques:

$$\sum T = I_j \cdot \ddot{\omega}_j + ms_j \times \dot{v}_j + \omega_j \times (I_j \cdot \omega_j) \quad (8)$$

Where ΣF are the external forces exerted on the body and ΣT are the external torques exerted on the body. And m_j is the mass of a particular body, ms_j is the vector distance from the chosen origin (that corresponds to the joint) to the center of mass of the body times the mass (without the mass the distance vector is called s_j), \dot{v}_j is the acceleration of the chosen origin, $\dot{\omega}_j$ is the angular acceleration of the body, ω_j is the angular velocity of the body and I is the Inertia of the body.

Translated to our particular body, the equations can be decomposed as follows:

Body Forces:

$$\sum F = F + m \cdot g \quad (9)$$

$$m \cdot \dot{v} = 0 \quad (10)$$

$$\dot{\omega} \times ms = \begin{bmatrix} 0 \\ 0 \\ \ddot{q} \end{bmatrix} \times m \cdot \begin{bmatrix} x \\ y \\ 0 \end{bmatrix} \quad (11)$$

$$\omega \times (\omega \times ms) = \begin{bmatrix} 0 \\ 0 \\ \dot{q} \end{bmatrix} \times \left(\begin{bmatrix} 0 \\ 0 \\ \dot{q} \end{bmatrix} \times m \cdot \begin{bmatrix} x \\ y \\ 0 \end{bmatrix} \right) \quad (12)$$

Where $g = -9.81 \text{ m.s}^{-2}$ is the gravity acceleration and the only unknown is F which makes reference to the reaction Forces of the first link against the ground.

And body 1 Torques:

$$\sum T = T + s \times (m \cdot g) \quad (13)$$

$$\sum T = T + \begin{bmatrix} x \\ y \\ 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ m \cdot g \\ 0 \end{bmatrix} \quad (14)$$

$$I \cdot \dot{\omega} = (m \cdot l^2) \cdot \begin{bmatrix} 0 \\ 0 \\ \ddot{q} \end{bmatrix} \quad (15)$$

$$ms \times \dot{v} = m \cdot \begin{bmatrix} x \\ y \\ 0 \end{bmatrix} \times \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ 0 \end{bmatrix} \quad (16)$$

$$\omega \times (I \cdot \omega) = 0 \quad (17)$$

Where, again, the only unknown is T which makes reference to the reaction Torque of the first link against the ground.

2.1.3 Kinematics planner

It is supposed that the motions are known at every moment. To be able to make that assumption, the motion of the system will be defined using a polynomial approach, relying on the initial and final conditions that can be chosen.

Knowing that, 6 initial conditions are known for our angle ($q(0), q(t_f), \dot{q}(0), \dot{q}(t_f), \ddot{q}(0), \ddot{q}(t_f)$) up to 6 equations can be obtained.

$$k_5 \cdot t^5 + k_2 \cdot t^4 + k_4 \cdot t^3 + k_3 \cdot t^2 + k_2 \cdot t + k_0 = q(t) \quad (18)$$

$$5 \cdot k_5 \cdot t^4 + 4 \cdot k_4 \cdot t^3 + 3 \cdot k_3 \cdot t^2 + 2 \cdot k_2 \cdot t + k_1 = \dot{q}(t) \quad (19)$$

$$20 \cdot k_5 \cdot t^3 + 12 \cdot k_4 \cdot t^2 + 6 \cdot k_3 \cdot t + 2 \cdot k_2 = \ddot{q}(t) \quad (20)$$

With all the k computed, the array of positions, velocities and accelerations needed will be generated.

2.1.4 Energy Computation

To compute the Energy of the system the Lagrange formulation will be applied:

$$E = \frac{1}{2} \cdot \begin{bmatrix} \dot{x} & \dot{y} & 0 & 0 & 0 & \dot{q} \end{bmatrix} \cdot \begin{bmatrix} m & \widehat{msi}^T \\ \widehat{msi} & I \end{bmatrix} \cdot \begin{bmatrix} \dot{x} \\ \dot{y} \\ 0 \\ 0 \\ 0 \\ \dot{q} \end{bmatrix} \quad (21)$$

$$U = m \cdot g \cdot y \quad (22)$$

For our body:

$$E = \frac{1}{2} \cdot \begin{bmatrix} \dot{x} & \dot{y} & 0 & 0 & 0 & \dot{q} \end{bmatrix} \cdot \begin{bmatrix} m & 0 & 0 & 0 & 0 & -m \cdot y \\ 0 & m & 0 & 0 & 0 & m \cdot x \\ 0 & 0 & m & m \cdot y & -m \cdot x & 0 \\ 0 & 0 & m \cdot y & 0 & 0 & 0 \\ 0 & 0 & -m \cdot x & 0 & 0 & 0 \\ -m \cdot y & m \cdot x & 0 & 0 & 0 & m \cdot l^2 \end{bmatrix} \cdot \begin{bmatrix} \dot{x} \\ \dot{y} \\ 0 \\ 0 \\ 0 \\ \dot{q} \end{bmatrix} \quad (23)$$

$$U = m \cdot g \cdot y \quad (24)$$

Where E is the kinetic E of the body and U is its potential energy. And \widehat{ms}_i is the skew-symmetric product of the ms_i vector.

2.1.5 Simulation and animation

To finish, an animation of the double pendulum that plots the force and torque of each joint is displayed. For this simulation, the final time is set to 1 second, thus the forces related to the accelerations are big enough to play a significant role compared to the force of the gravity.

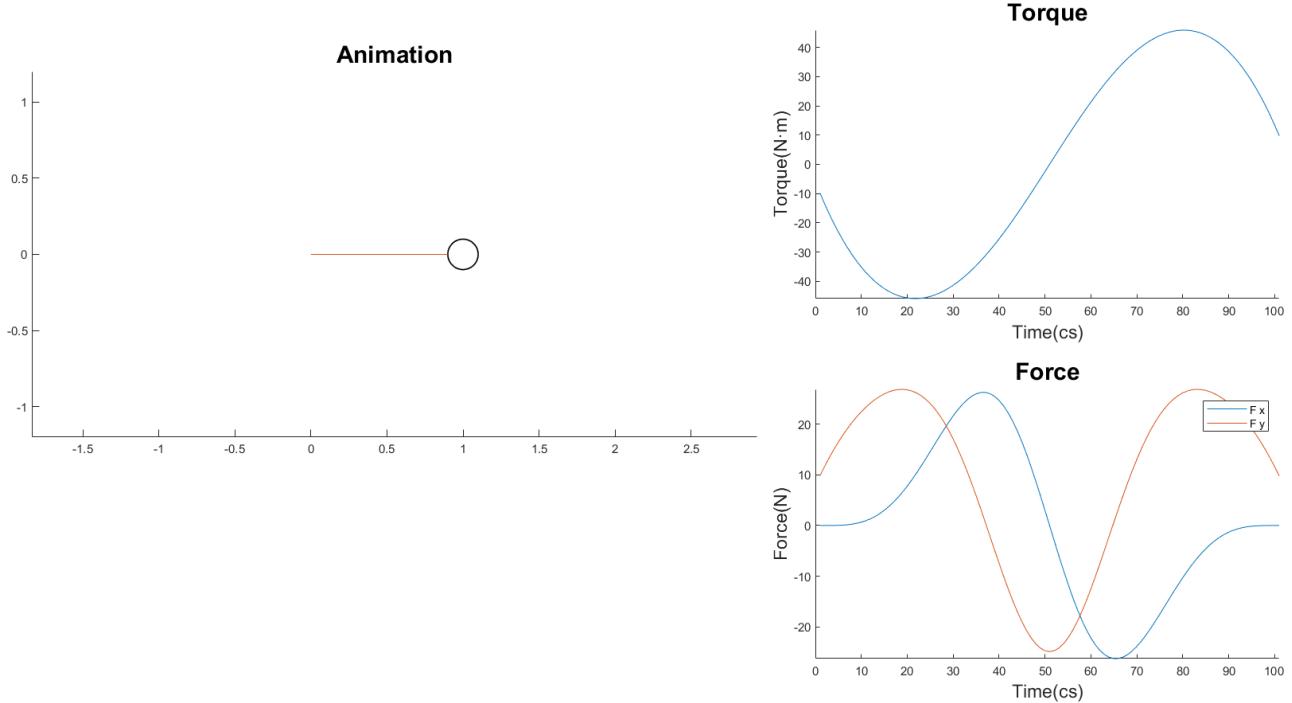


Figure 2: Single pendulum animation and reactions

2.2 Double pendulum

An inverted double pendulum pendulum is now considered: a point mass m_1 attached to a rotational joint by a mass-less rod and another point mass m_2 attached to the previous point mass by a another mass-less rod:

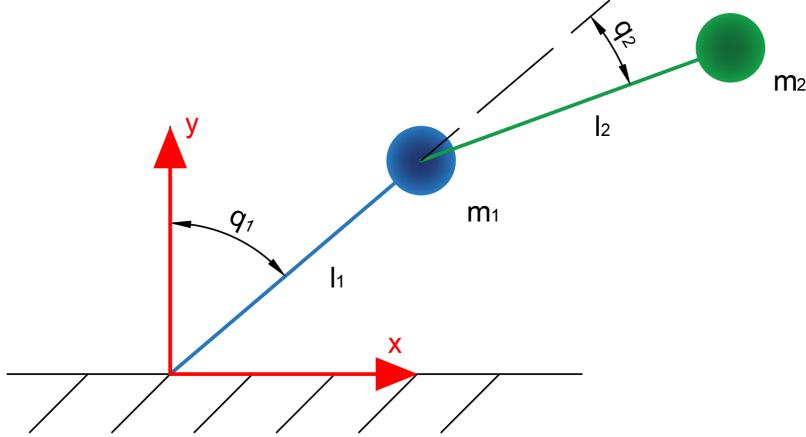


Figure 3: Double pendulum scheme

Where m_1 and m_2 are the masses of each pendulum, l_1 and l_2 is the length of the link, and q_1 is the angle between the link and the y global axis while q_2 is the angle between link 2 and link 1.

2.2.1 Kinematic model

The equations of the position of the mass depending on the angle position are computed.

$$x_1 = -l_1 \sin(q_1) \quad (25)$$

$$y_1 = l_1 \cos(q_1) \quad (26)$$

$$x_2 = -l_1 \sin(q_1) - l_2 \sin(q_1 + q_2) \quad (27)$$

$$y_2 = l_1 \cos(q_1) + l_2 \cos(q_1 + q_2) \quad (28)$$

Where x_1 , y_1 , x_2 and y_2 are the position of the masses m_1 and m_2 respectively, with respect to the origin.

The equations of the velocity of the masses depending on the angle positions and angular velocities are found with the same method as before by deriving the four previous equations.

$$\dot{x}_1 = -l_1 \cos(q_1) \dot{q}_1 \quad (29)$$

$$\dot{y}_1 = -l_1 \sin(q_1) \dot{q}_1 \quad (30)$$

$$\dot{x}_2 = -l_1 \cos(q_1) \dot{q}_1 - l_2 \cos(q_1 + q_2) (\dot{q}_1 + \dot{q}_2) \quad (31)$$

$$\dot{y}_2 = -l_1 \sin(q_1) \dot{q}_1 - l_2 \sin(q_1 + q_2) (\dot{q}_1 + \dot{q}_2) \quad (32)$$

Where \dot{x}_1 , \dot{y}_1 , \dot{x}_2 and \dot{y}_2 are the velocity of the masses m_1 and m_2 respectively, with respect to the origin.

Finally, the acceleration of the masses depending on the angle positions, angular velocities and angular accelerations are also obtained by deriving the velocity equations.

$$\ddot{x}_1 = l_1 \sin(q_1) \dot{q}_1^2 - l_1 \cos(q_1) \ddot{q}_1 \quad (33)$$

$$\ddot{y}_1 = -l_1 \cos(q_1) \dot{q}_1^2 - l_1 \sin(q_1) \ddot{q}_1 \quad (34)$$

$$\ddot{x}_2 = -l_1 \cos(q_1) \ddot{q}_1 + l_1 \sin(q_1) \dot{q}_1^2 - l_2 \cos(q_1 + q_2) (\ddot{q}_1 + \ddot{q}_2) + l_2 \sin(q_1 + q_2) (\dot{q}_1 + \dot{q}_2)^2 \quad (35)$$

$$\ddot{y}_2 = -l_1 \sin(q_1) \ddot{q}_1 + l_1 \sin(q_1) \dot{q}_1^2 - l_2 \sin(q_1 + q_2) (\ddot{q}_1 + \ddot{q}_2) + l_2 \cos(q_1 + q_2) (\dot{q}_1 + \dot{q}_2)^2 \quad (36)$$

Where \ddot{x}_1 , \ddot{y}_1 , \ddot{x}_2 and \ddot{y}_2 are the acceleration of the masses m_1 and m_2 respectively, with respect to the origin.

2.2.2 Dynamic model

To solve the dynamic model, the Newton-Euler approach is used again beginning with the last body(body 2):

Body 2 Forces:

$$\sum F_2 = F_2 + m_2 \cdot g \quad (37)$$

$$m_2 \cdot \dot{v}_2 = m_2 \cdot \begin{bmatrix} \ddot{x}_1 \\ \ddot{y}_1 \\ 0 \end{bmatrix} \quad (38)$$

$$\dot{\omega}_2 \times ms_2 = \begin{bmatrix} 0 \\ 0 \\ (\dot{q}_1 + \dot{q}_2) \end{bmatrix} \times m \cdot \begin{bmatrix} (x_2 - x_1) \\ (y_2 - y_1) \\ 0 \end{bmatrix} \quad (39)$$

$$\omega_2 \times (\dot{\omega}_2 \times ms_2) = \begin{bmatrix} 0 \\ 0 \\ (\dot{q}_1 + \dot{q}_2) \end{bmatrix} \times \left(\begin{bmatrix} 0 \\ 0 \\ (\dot{q}_1 + \dot{q}_2) \end{bmatrix} \times m \cdot \begin{bmatrix} (x_2 - x_1) \\ (y_2 - y_1) \\ 0 \end{bmatrix} \right) \quad (40)$$

Where the only unknown is F_2 which makes reference to the reaction forces of the first link against the second one.

And body 2 Torques:

$$\sum T_2 = T_2 + s_2 \times (m_2 \cdot g) = T_2 + \begin{bmatrix} (x_2 - x_1) \\ (y_2 - y_1) \\ 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ m_2 \cdot g \\ 0 \end{bmatrix} \quad (41)$$

$$I_2 \cdot \dot{\omega}_2 = (m_2 \cdot l_2^2) \cdot \begin{bmatrix} 0 \\ 0 \\ (\ddot{q}_1 + \ddot{q}_2) \end{bmatrix} \quad (42)$$

$$ms_2 \times \dot{v}_2 = m \begin{bmatrix} (x_2 - x_1) \\ (y_2 - y_1) \\ 0 \end{bmatrix} \times \begin{bmatrix} (\ddot{x}_2 - \ddot{x}_1) \\ (\ddot{y}_2 - \ddot{y}_1) \\ 0 \end{bmatrix} \quad (43)$$

$$\omega_2 \times (I_2 \cdot \omega_2) = 0 \quad (44)$$

Where, again, the only unknown is T_2 which makes reference to the reaction Torque of the first link against the second one.

Following with body 1 Forces where body 2 applied the reciprocal forces:

$$\sum F_1 = F_1 + m_1 \cdot g - F_2 \quad (45)$$

$$m_1 \cdot \dot{v}_1 = 0 \quad (46)$$

$$\dot{\omega}_1 \times ms_1 = \begin{bmatrix} 0 \\ 0 \\ \ddot{q}_1 \end{bmatrix} \times m \begin{bmatrix} x_1 \\ y_1 \\ 0 \end{bmatrix} \quad (47)$$

$$\omega_1 \times (\omega_1 \times ms_1) = \begin{bmatrix} 0 \\ 0 \\ \dot{q}_1 \end{bmatrix} \times \left(\begin{bmatrix} 0 \\ 0 \\ \ddot{q}_1 \end{bmatrix} \times m \begin{bmatrix} (x_1) \\ (y_1) \\ 0 \end{bmatrix} \right) \quad (48)$$

Where the only unknown is F_1 which makes reference to the reaction Forces of the first link against the ground.

And body 1 Torques:

$$\sum T_1 = T_1 + s_1 \times (m_1 \cdot g) - T_2 - s_1 \times F_2 = T_1 + \begin{bmatrix} x_1 \\ y_1 \\ 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ m_2 \cdot g \\ 0 \end{bmatrix} - T_2 - \begin{bmatrix} x_1 \\ y_1 \\ 0 \end{bmatrix} \times F_2 \quad (49)$$

$$I_1 \cdot \dot{\omega}_1 = (m_1 \cdot l_1^2) \cdot \begin{bmatrix} 0 \\ 0 \\ \ddot{q}_1 \end{bmatrix} \quad (50)$$

$$ms_1 \times \dot{v}_1 = m \begin{bmatrix} x_1 \\ y_1 \\ 0 \end{bmatrix} \times \begin{bmatrix} \ddot{x}_1 \\ \ddot{y}_1 \\ 0 \end{bmatrix} \quad (51)$$

$$\omega_1 \times (I_1 \cdot \omega_1) = 0 \quad (52)$$

Where, again, the only unknown is T_1 which makes reference to the reaction Torque of the first link against the ground.

2.2.3 Kinematics planner

It is supposed that the motions are known at every moment. To be able to make that assumption, it will be defined the motion of the system using a polynomial approach, relying on the initial and final conditions.

Knowing that 6 initial conditions are known for each angle ($q_i(0)$, $q_i(t_f)$, $\dot{q}_i(0)$, $\dot{q}_i(t_f)$, $\ddot{q}_i(0)$, $\ddot{q}_i(t_f)$) up to 6 equations for each articulation can be obtained.

$$k_{5i} \cdot t^5 + k_{4i} \cdot t^4 + k_{3i} \cdot t^3 + k_{2i} \cdot t^2 + k_{1i} \cdot t + k_{0i} = q_i(t) \quad (53)$$

$$5 \cdot k_{5i} \cdot t^4 + 4 \cdot k_{4i} \cdot t^3 + 3 \cdot k_{3i} \cdot t^2 + 2 \cdot k_{2i} \cdot t + k_{1i} = \dot{q}_i(t) \quad (54)$$

$$20 \cdot k_{5i} \cdot t^3 + 12 \cdot k_{4i} \cdot t^2 + 6 \cdot k_{3i} \cdot t + 2 \cdot k_{2i} = \ddot{q}_i(t) \quad (55)$$

With all the k_i computed, the array of positions, velocities and accelerations needed can be generated.

2.2.4 Energy Computation

To compute the Energy of the system, the Lagrange formulation will be used:

$$E = \frac{1}{2} \cdot \begin{bmatrix} \dot{x} & \dot{y} & 0 & 0 & 0 & \dot{q} \end{bmatrix} \cdot \begin{bmatrix} m & \widehat{msi}u^T \\ \widehat{msi} & I \end{bmatrix} \cdot \begin{bmatrix} \dot{x} \\ \dot{y} \\ 0 \\ 0 \\ 0 \\ \dot{q} \end{bmatrix} \quad (56)$$

$$U = m \cdot g \cdot y \quad (57)$$

Where E is the kinetic E of the body and U is its potential energy. And msi is the skew-symmetric product of the center of masses with respect of the chosen origin of the body

For body 1:

$$E_1 = \frac{1}{2} \cdot \begin{bmatrix} \dot{x}_1 & \dot{y}_1 & 0 & 0 & 0 & \dot{q}_1 \end{bmatrix} \cdot \begin{bmatrix} m_1 & 0 & 0 & 0 & 0 & -m_1 \cdot y_1 \\ 0 & m_1 & 0 & 0 & 0 & m_1 \cdot x_1 \\ 0 & 0 & m_1 & m_1 \cdot y_1 & -m_1 \cdot x_1 & 0 \\ 0 & 0 & m_1 \cdot y_1 & 0 & 0 & 0 \\ 0 & 0 & -m_1 \cdot x_1 & 0 & 0 & 0 \\ -m_1 \cdot y_1 & m_1 \cdot x_1 & 0 & 0 & 0 & m_1 \cdot l_1^2 \end{bmatrix} \cdot \begin{bmatrix} \dot{x}_1 \\ \dot{y}_1 \\ 0 \\ 0 \\ 0 \\ \dot{q}_1 \end{bmatrix} \quad (58)$$

$$U_1 = m_1 \cdot g \cdot y_1 \quad (59)$$

For body 2:

$$E_2 = \frac{1}{2} \cdot \begin{bmatrix} \dot{x}_2 & \dot{y}_2 & 0 & 0 & 0 & (\dot{q}_1 + \dot{q}_2) \end{bmatrix} \cdot \begin{bmatrix} m_2 & 0 & 0 & 0 & 0 & -m_2 \cdot y_2 \\ 0 & m_2 & 0 & 0 & 0 & m_2 \cdot x_2 \\ 0 & 0 & m_2 & m_2 \cdot y_2 & -m_2 \cdot x_2 & 0 \\ 0 & 0 & m_2 \cdot y_2 & 0 & 0 & 0 \\ 0 & 0 & -m_2 \cdot x_2 & 0 & 0 & 0 \\ -m_2 \cdot y_2 & m_2 \cdot x_2 & 0 & 0 & 0 & m_2 \cdot l_2^2 \end{bmatrix} \cdot \begin{bmatrix} \dot{x}_2 \\ \dot{y}_2 \\ 0 \\ 0 \\ 0 \\ (\dot{q}_1 + \dot{q}_2) \end{bmatrix}. \quad (60)$$

$$U_2 = m_2 \cdot g \cdot y_2 \quad (61)$$

2.2.5 Simulation and animation

To finish, we display an animation of the double pendulum that plots the force and torque of each joint. For this particular simulation, we have set the final time to be 1 second, so the forces related to the accelerations are big enough to play a significant role compared to the force of the gravity.

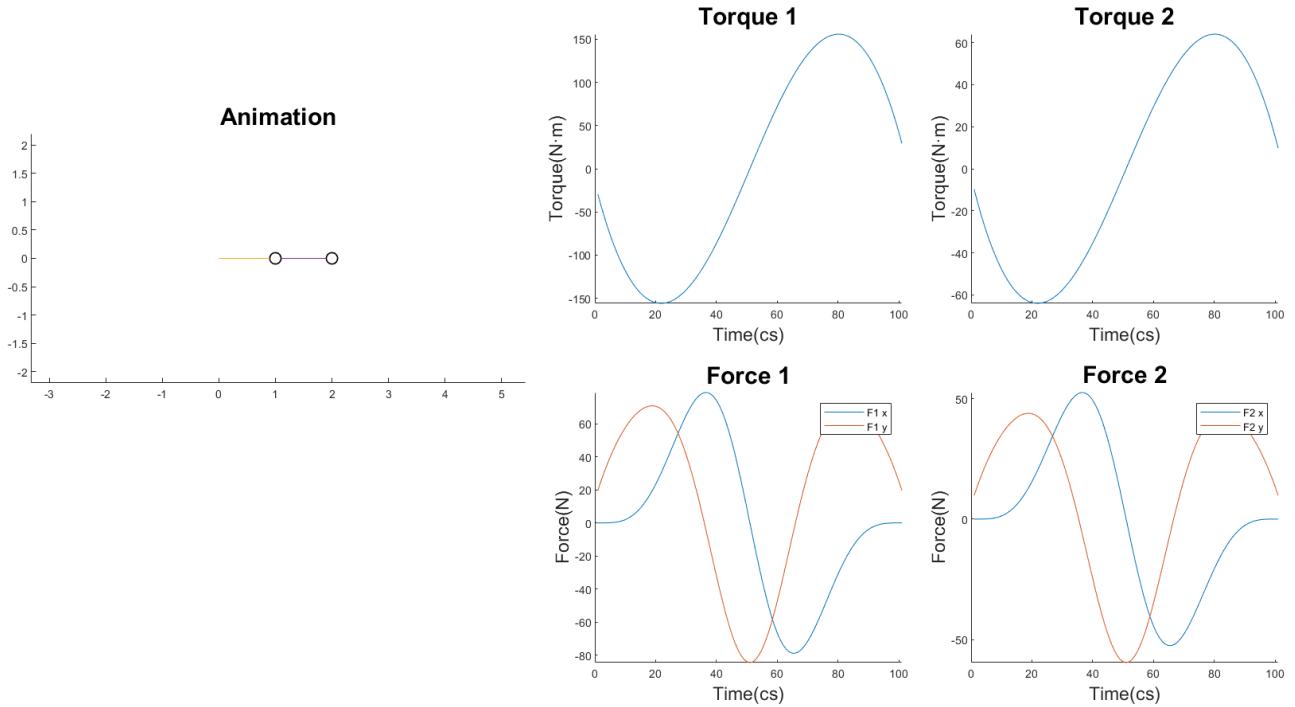


Figure 4: Double pendulum animation and reactions

3. Biomechanical model

To apply dynamic equations on a human body, a biomechanical model must be applied. The human body is modeled by set of rigid bodies that are usually linked together with rotational or spherical joints. Several biomechanical models exist, the one used here is the modified Hanavan one presented in the Figure 5. In this model, the hands and the head are represented with ellipsoidal of revolution solids as shown in the Figure 6. The arms are separated in two parts (the upper-arms and the forearms), the legs are separated in two parts (the thighs and the shanks) and the trunk is separated in three parts. Each of these parts are modeled with elliptical solids as presented in the Figure 7.

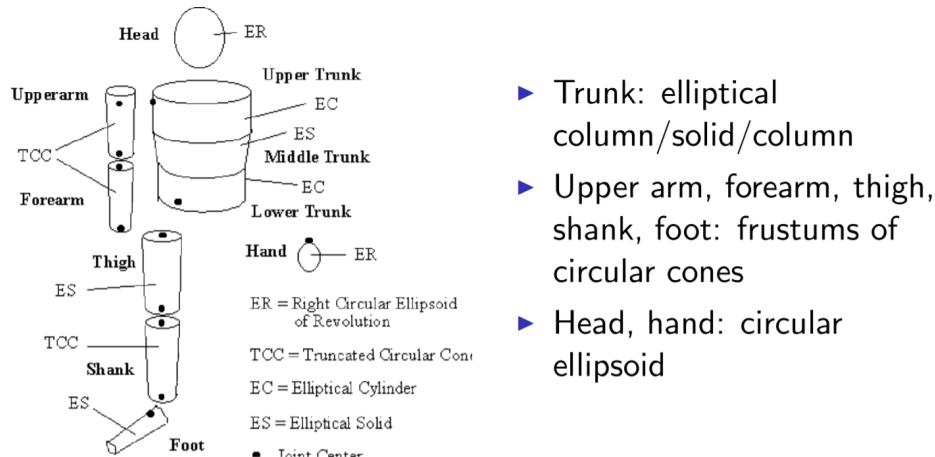


Figure 5: Modified Hanavan biomechanical model

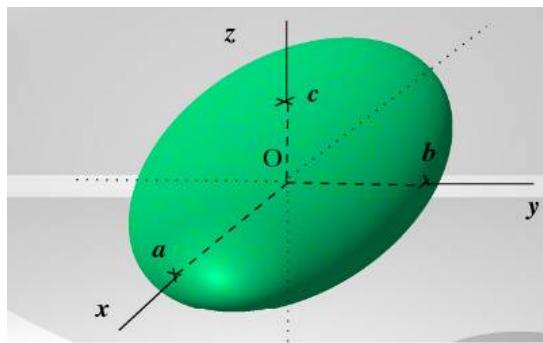


Figure 6: Geometry of an ellipsoid of revolution
Picture from <http://jeux-et-mathematiques.davalan.org/calc/ellipsoid/index.html> (03/12/2020)

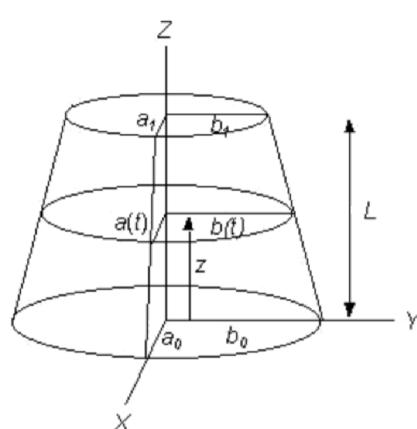


Figure 7: Geometry of an elliptical solid

The ellipsoidal solid is defined through the dimension a, b and c visible in the picture. With this form of a solid, the volume, the center of mass (CoM) and the inertia matrix (which is diagonal because the solid has at least 2 planes of symmetry, 3 to be exact) by the computation:

Volume	$\frac{4\pi}{3} \cdot a \cdot b \cdot c$
CoM position	0
I _{xx}	$\frac{1}{5}m(b^2 + c^2)$
I _{yy}	$\frac{1}{5}m(a^2 + c^2)$
I _{zz}	$\frac{1}{5}m(a^2 + b^2)$

The elliptical solid is defined through the dimension L , a_0 , b_0 , a_1 , b_1 visible in the picture. It is supposed that the evolution from a_0 to a_1 and from b_0 to b_1 along the z axis is linear ($a(z) = a_0 + (a_1 - a_0) \cdot \frac{z}{L}$, $b(z) = b_0 + (b_1 - b_0) \cdot \frac{z}{L}$). With this form of a solid, the volume, the center of mass (CoM) and the inertia matrix (which is diagonal because the solid has at least 2 planes of symmetry, 2 to be exact) by the computation:

Volume	$\pi \cdot L \cdot A_1^{ab}$
CoM position	$L \cdot \frac{A_2^{ab}}{A_1^{ab}}$
I _{xx}	$\frac{1}{4}m \frac{A_4^{abbb}}{A_1^{ab}} + mL^2 \frac{A_3^{ab}}{A_1^{ab}} - m \left(L \frac{A_2^{ab}}{A_1^{ab}} \right)^2$
I _{yy}	$\frac{1}{4}m \frac{A_4^{aabb}}{A_1^{ab}} + mL^2 \frac{A_3^{ab}}{A_1^{ab}} - m \left(L \frac{A_2^{ab}}{A_1^{ab}} \right)^2$
I _{zz}	$\frac{1}{4}m \frac{A_4^{aabb} + A_4^{abbb}}{A_1^{ab}}$

Where the coefficients A are defined with the following equations:

$$\begin{aligned}
 A_1^{ab} &= \frac{B_1^{ab}}{3} + \frac{B_2^{ab}}{2} + B_3^{ab} \\
 A_2^{ab} &= \frac{B_1^{ab}}{4} + \frac{B_2^{ab}}{3} + \frac{B_3^{ab}}{2} \\
 A_3^{ab} &= \frac{B_1^{ab}}{5} + \frac{B_2^{ab}}{4} + \frac{B_3^{ab}}{3} \\
 A_4^{abcd} &= \frac{B_4^{abcd}}{5} + \frac{B_5^{abcd}}{4} + \frac{B_6^{abcd}}{3} + \frac{B_7^{abcd}}{2} + B_8^{abcd}
 \end{aligned} \tag{62}$$

Where the coefficients B are defined with the following equations:

$$\begin{aligned}
 B_1^{ab} &= (a_1 - a_0)(b_1 - b_0) \\
 B_2^{ab} &= a_0(b_1 - b_0) + b_0(a_1 - a_0) \\
 B_3^{ab} &= a_0b_0 \\
 B_4^{abcd} &= (a_1 - a_0)(b_1 - b_0)(c_1 - c_0)(d_1 - d_0) \\
 B_5^{abcd} &= a_0(b_1 - b_0)(c_1 - c_0)(d_1 - d_0) + b_0(a_1 - a_0)(c_1 - c_0)(d_1 - d_0) \\
 &\quad + c_0(a_1 - a_0)(b_1 - b_0)(d_1 - d_0) + d_0(a_1 - a_0)(b_1 - b_0)(c_1 - c_0) \\
 B_6^{abcd} &= a_0b_0(c_1 - c_0)(d_1 - d_0) + a_0c_0(b_1 - b_0)(d_1 - d_0) + a_0d_0(b_1 - b_0)(c_1 - c_0) \\
 &\quad + b_0c_0(a_1 - a_0)(d_1 - d_0) + b_0d_0(a_1 - a_0)(c_1 - c_0) + c_0d_0(a_1 - a_0)(b_1 - b_0) \\
 B_7^{abcd} &= b_0c_0d_0(a_1 - a_0) + a_0c_0d_0(b_1 - b_0) + a_0b_0d_0(c_1 - c_0) + a_0b_0c_0(d_1 - d_0) \\
 B_8^{abcd} &= a_0b_0c_0d_0
 \end{aligned} \tag{63}$$

To get the inertia matrix at the center of mass for elliptical solids, the value $m.z_{CoM}^2$ must be subtracted to I_{xx} and I_{yy}, where z_{CoM} is the vertical length from the origin to the center of mass of the elliptical solid (For further explanations, see equation 75 in the Newton-Euler computation).

The computation of all B and A factors has been defined through some functions in the codes (for example $B=B_ab1(a0,a1,b0,b1)$) that are used in the function $I=Inertia_ES_group(segment)$ which compute the inertia matrix of elliptical solid. In the same way, the inertia matrix of ellipsoidal solid is computed with the function $I=Inertia_SE_group(segment)$. The volume of the elliptical solid is computed with the function $V=Volume_ES_group(segment)$ and the one of ellipsoidal solid by $V=Volume_SE_group(segment)$ (ES stands for Elliptical Solid and SE for Semi Ellipsoid, which is named incorrectly for this last). Finally, the function $CoM=CoM_ES_group(segment)$ compute the position of the center of mass position for the elliptical solids.

The input of these functions is a Matlab structure containing the geometrical parameters (a , b and c or a_0 , b_0 , a_1 , b_1 and L). Also, one has to note that the reference frame of ellipsoidal solid is the center of the solid (thus the center of mass) and the one for elliptical solid is the center of the base ellipse (of dimension a_0 , b_0).

No	Parameter	No	Parameter
1	L: Hand	21	C: Toe
2	L: Wrist to Knuckle	22	C: Ankle
3	L: Forearm	23	C: Shank
4	L: Upper arm	24	C: Knee
5	L: Elbow to Acromion	25	C: Upper Thigh
6	L: Foot	26	C: Head
7	L: Shank	27	C: Chest
8	L: Thigh	28	C: Xyphion Level
9	L: Head	29	C: Omphalion Level
10	L: Upper Trunk	30	C: Buttock
11	L: Xyphion to Acromion Level	31	W: Hand
12	L: Middle Trunk	32	W: Wrist
13	L: Lower Trunk	33	W: Foot
14	C: Fist	34	W: Toe
15	C: Wrist	35	D: Hip
16	C: Forearm	36	W: Chest
17	C: Elbow	37	W: Xyphion Level
18	C: Axillary Arm	38	W: Omphalion Level
19	C: Foot	39	W: Coxae
20	C: Ball of Foot	40	L: Xyphion Level to Chin/Neck Intersection
41	L: Hip to Chin/Neck Intersection = P12 + P13 + P40		

L: Length; C: Circumference; W: Width; D: Depth

Figure 8: Hanavan parameters

Therefore, to apply the modified Hanavan biomechanical model to a human body, one must determine adapted dimension of each modeled body to the subject. For that, one has to determine the Hanavan parameters presented in the Figure 8 by measuring some length, circumference, width and depth. A tape measure can be used to perform the measurements. Once these parameters are found, the geometrical parameter of each body can be found using the formulas of the Figure 9.

Segment	Shape	Group	Arguments
Hand	ER	SE	$a = b = \frac{P_{14}}{2\pi}, c = \frac{P_2}{2}$
Forearm	TCC	ES	$a_0 = b_0 = \frac{P_{17}}{2\pi}, a_1 = b_1 = \frac{P_{15}}{\pi}, L = P_3$
Upperarm	TCC	ES	$a_0 = b_0 = \frac{P_{18}}{2\pi}, a_1 = b_1 = \frac{P_{17}}{\pi}, L = P_5$
Foot	ES Circ. Base	ES	$a_0 = b_0 = \frac{P_{19}}{2\pi}, a_1 = \frac{P_{33}+P_{34}}{4}, b_1 = \frac{P_{20}+P_{21}}{2\pi}, L = P_6$
Shank	TCC	ES	$a_0 = b_0 = \frac{P_{24}}{2\pi}, a_1 = b_1 = \frac{P_{22}}{2\pi}, L = P_7$
Thigh	ES Circ. Top	ES	$b_0 = \frac{P_{35}}{2}, a_0 = \frac{P_{25}}{\pi} - b_0, a_1 = b_1 = \frac{P_{24}}{2\pi}, L = P_8$
Head	ER	SE	$a = b = \frac{P_{26}}{2\pi}, c = \frac{P_9}{2}$
U Trunk	EC	ES	$a_0 = a_1 = \frac{P_{36}+P_{37}}{4}, b_0 = b_1 = \frac{P_{27}+P_{28}}{2\pi} - a_0, L = P_{11}$
M Trunk	ES	ES	$a_0 = \frac{P_{37}}{2}, a_1 = \frac{P_{38}}{2}, L = P_{12}, b_0 = \frac{P_{28}}{\pi} - a_0, b_1 = \frac{P_{29}}{\pi} - a_1$
L Trunk	EC	ES	$a_0 = a_1 = \frac{P_{38}+P_{39}}{4}, L = P_{13}, b_0 = b_1 = \frac{P_{29}+P_{30}}{2\pi} - a_0$

EC: Elliptical Column, ER: Ellipsoid of Revolution, ES: Elliptical Solid, SE:

Semi-Ellipsoid, TCC Truncated Circular Cone

Figure 9: Hanavan solids geometry

4. Data acquisition

4.1 Capture system

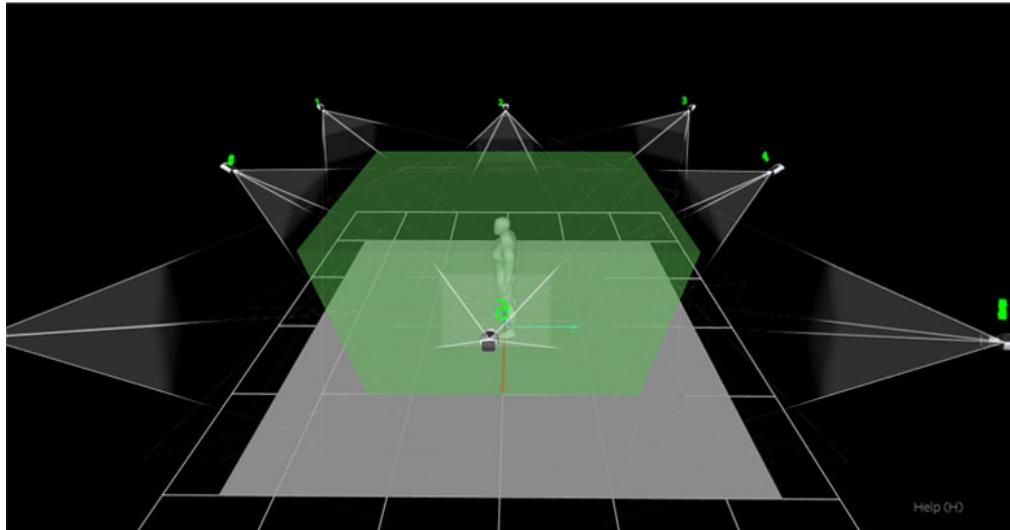


Figure 10: Example of the cameras positions

Picture from <https://mocap.reallusion.com/iclone-motion-live-mocap/qualisys.html> (03/12/2020)

The motion capture system is composed of a set of multiple infrared cameras put in a way that they are able to look at a certain space from a lot of different angles and positions as shown in the picture Figure 10. This camera will emit infrared rays and observe their reflection by a set of markers put on the subject. These markers can be directly put on the skin or on a suit that the subject wears. The marker suits used for this motion capture is presented in the pictures of the Figure 11.

This system allows to observe the evolution of the markers in the space and by associating the marker to their position in the body (and by associating the human body to a rigid articulated biomechanical model), the system is able to present the evolution of the biomechanical model in the space over time.

A force plate has also been added to the ground, this plate allows to get the forces along the three axes of the space and the torques around the three axes of the space at one point of the ground.

Once the subject is equipped with the markers, he can perform some motion to acquire. The motion capture system and the force plate software working separately, they must be synchronized by hand (or voice) for every motion.



Figure 11: Marker suit

4.2 Data provided

Due to the sanitary situation, the measurement couldn't be performed, thus data of previous year students has been used.

4.2.1 Hanavan parameters

First, the measured Hanavan parameters are presented in the Figure 12. On these data, the Hanavan parameter has been measured by different persons. Thereafter, the mean value of the measurements, used for our model has been computed by:

$$\bar{X} = \frac{1}{N} \sum_{i=1}^N X_i \quad (64)$$

Then, to know if a mean value of a parameter is really representative of the measured person, the standard deviation has been calculated:

$$\sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (X_i - \bar{X})^2} \quad (65)$$

	A	B	C	D	E	F	G	H	I
1	No	eBona_Manca	Labolani_Lottero_Olivi	Franklin_Deena	Tasheed_Caterin.	Mean	Standard Deviation	Mean_in_meters	
2	1	19.1	20.3	20.5	18	19.1	1.019803903	0.194	
3	2	9	9.4	11	10.5	10	0.809320703	0.1	
4	3	27.2	25.1	29	32	28.5	2.532390175	0.2836	
5	4	35	31.5	33	33	32.1	3.3292	1.325518766	
6	5	37.4	35.8	33.5	32	35.5	3.484	2.107842499	
7	6	27	27	27	27.5	26.4	2.698	0.369871774	
8	7	32	40.3	38	49	40.3	3.932	6.106306214	
9	8	38	42.2	41	44.5	23.6	3.306	5.784289066	
10	9	19.5	22.6	23.5	22.4	21.8	2.196	1.504327092	
11	10	20	20.7	22	17.5	29.1	2.186	4.366119559	
12	11	29	28.6	32	18	22	2.592	5.7388640954	
13	12	8	8.3	8	30.5	9	12.76	3.92537153	
14	13	16	17.6	20	13.5	14.8	1.638	2.526262061	
15	14	28.4	27.3	27.5	29.4	28.3	2.818	0.834865259	
16	15	17.6	17.2	19	17	17	17.56	0.641427359	
17	16	24.4	26.4	23.5	27.1	26.7	2.562	1.577022511	
18	17	25.5	26.3	25.5	27	26.2	2.61	0.628490254	
19	18	32	30.9	33	31.3	30.95	3.163	0.862892972	
20	19	22.3	25.2	26	23	24.7	2.424	1.543696861	
21	20	23.8	25.5	26.5	18.5	21.45	2.315	3.228002478	
22	21	8	8.6	9	8	8.1	0.834	0.444971909	
23	22	22.3	23.3	23.5	26.8	27.4	2.466	2.283193509	
24	23	38	28.9	37.5	35.7	34.5	3.492	3.645819524	
25	24	36	38	46	38	36.5	3.89	4.068163121	
26	25	44	48	35.5	53.2	53.1	46.76	7.375838935	
27	26	58	58	59	61.5	39.1	55.12	9.069013177	
28	27	98	99.2	100	96.8	96.9	98.18	1.407835218	
29	28	86	88.5	87.5	96.5	86.9	89.48	5.123670559	
30	29	76.7	81	77.2	80.7	83.2	79.76	2.746452257	
31	30	94	93.8	93.5	96.5	93.3	94.22	1.302689526	
32	31	9.4	10.6	9.8	10.4	11.7	10.38	0.878635305	
33	32	6.4	6.7	6.8	8.8	6.3	7	1.027131929	
34	33	9	7.8	9	11	10.3	9.42	1.249739984	
35	34	2.5	3.4	3	2.9	2.3	2.82	0.432434966	
36	35	15	17.7	16	21.5	21.5	18.34	3.041874422	
37	36	33	32	32.5	35.5	30.1	32.62	1.948589233	
38	37	29	29.5	28.9	33.8	28.6	29.96	2.170944495	
39	38	29.5	29.3	30	35.5	31.8	31.22	2.587856256	
40	39	30	31.5	31	33.5	27.5	30.7	2.196588264	
41	40	27	31.5	29	28.5	34.6	30.12	2.982783935	
42	41	51	57.4	57	65.5	57.25	5.952800461	0.57725	

Figure 12: Mesured Hanavan parameters

In this Figure 12, it can be seen that some parameter has bigger value of standard deviation (like P7, P8, P11, P12, P25, P26, P28 and P51 with a standard deviation above 5cm). These high standard deviations indicate that the mean value could not be representative of the subject and can be the cause of error in the following work. This high standard deviation is due to the fact that there is no indicating way to measure these parameters, thus the lengths measured on the body are different between the persons who performed the measurement because of the choice they made.

4.2.2 Motion and force plate data

Several motions have been performed during the last year:

- quick jump, med jump, max jump, jump feet up (corresponding to different types of jump)
- slow arm, medium arm, fast arm (an arm motion with different velocities)
- slow kick, medium kick, fast kick (a leg motion with different velocities)

- slow kick arm, medium kick arm , fast kick arm (a leg and arm motion with different velocities)
- slow sit (the motion of a person sitting)
- custom and custom 2 (a motion where the person balance his body forward on one leg or the other)

The data provided by the force plate is presented under a csv file, each cell containing:

- the sample number,
- the time instant,
- the force along x,y and z
- the torques around x, y and z (from a point)
- the x and y coordinates of the center of pressure.

In the folder **Reading files** a script called **read_csv.m** allows to get the data from the csv file under a matlab structure array containing the evolution of these data over time.

The data provided by the motion capture are drf files (text files). This data is divided under line. After the line indicating the time step (line beginning by ts), two important lines follow: one line containing a the data for 17 frames (line beginning by 6d 17) and the other one a line containing the data of 20 frames (line beginning by 6dj) corresponding to joints data, these points data corresponding to the time instant.

For each of these lines, the data of a frame is presented with a first list containing the point index (of the form [0 i] with i the point index), following with a list of 6 value (corresponding to the position and the orientation of the frame) and a list of 9 unknown values. Each data of a point following the previous one in the line.

In the folder **Reading files**, a script **read_drf.m** allows to get these data in a matlab structure array containing the time vector and the corresponding position and orientation for each frame over time by reading character by character the drf files.

5. Use of the data

5.1 Construction of the biomechanical model bodies

From the Hanavan parameters, the geometric parameters of each body can be determined following the formulas Figure 9. In the folder **Hanavan model**, the script **Hanavan.m** uses these parameters values to compute the geometrical values of each body.

The Hanavan parameters are also used to determine the mass of each link following the Hanavan mass prediction formulas presented in the Figure 13 (one should be aware that the formulas used in the Hanavan parameters are in cm). From the force plate data for slow motion where the subject is not moving a lot, the force along the vertical axis (z-axis) allows to estimate the total mass of the subject about 80 kg. Thus, applying the formulas with this total mass value and with these parameters value, the Hanavan mass predictions lead to a total mass of the subject about 86.61 kg. This computed value of the mass can differ from the real value because of the value of the Hanavan parameter that would not be really representative of the body as said in the Section 4.2.1. Other errors occur like the computed mass of the forearm (6.16 kg) which is three times superior to the mass of the upper-arm (2.11 kg), these errors will be important in high dynamics motion analysis.

Once these values are determined, the functions presented in the Section 3 are used to find the volume, the position of the center of mass and the inertia of each body. At the end of the program, all the computed characteristics are stacked in a matlab structure array named as the corresponding link. For example, the structure *Hand* has the following fields:

- a, b, c for the geometric values (it is $a0, a1, b0, b1$ and L for an elliptical solid),
- m for the mass,

- *inertia* for the inertia matrix,
- *volume* for the volume,
- *CoM* for the position of the center of mass,
- *name* for the name of the body, "Hand" for this example
- *shape*, "ER" for this example,
- *group*, "SE" for this example.

Segment	Prediction Equation
Hand	$m = 0.038*P_{15} + 0.080*P_{32} - 0.660$
Forearm	$m = 0.081*M + 0.052*P_{16} - 1.650$
Upper arm	$m = 0.007*M + 0.092*P_{18} + 0.050*P_5 - 3.101$
Foot	$m = 0.003*M + 0.048*P_{22} + 0.027*P_6 - 0.869$
Shank	$m = 0.135*P_{23} - 1.318$
Thigh	$m = 0.074*M + 0.138*P_{25} - 4.641$
Head	$m = 0.104*P_{26} + 0.015*M - 2.189$
Whole trunk	$m_{wt} = 0.349*M + 0.423*P_{41} + 0.229*P_{27} - 35.460$
Upper trunk	$m = 0.92*V_{ut}*sf$
Middle trunk	$m = 1.01*V_{mt}*sf$
Lower trunk	$m = 1.01*V_{lt}*sf$

M : Whole body mass; P_i : Anthropometric parameter i ; V : volume.

$$sf = \text{Scaling factor} = \frac{m_{wt}}{0.92 * V_{ut} + 1.01 * (V_{mt} + V_{lt})}$$

Figure 13: Hanavan mass prediction

5.2 Use of the motion data

The data used for our work was the set of 20 frames, the positions of these frames on the body are presented in the Figure 14. Each frame is associated to a body, but certain frames belong to the same body. Indeed for 20 frames, 16 bodies have been defined previously, thus not all frames are used, even less than that because the lower trunk and the middle trunk are considered as one body (leading to a total of 15 bodies).

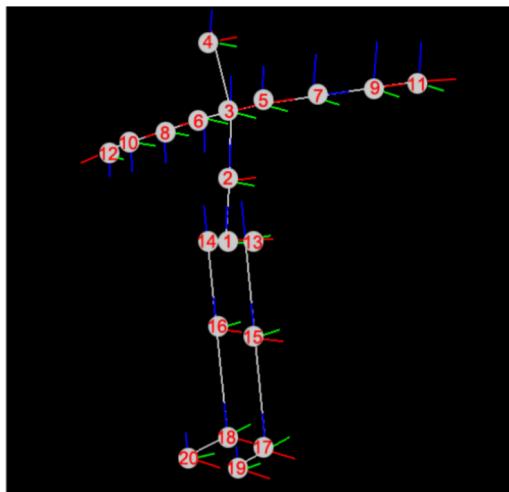


Figure 14: Position of the 20 frames

For each time step, one has the absolute position of the frame in the space (expressed in mm) and three orientations (in degree). These orientations are Euler angles of sequence XYZ and depending on the frame, they correspond to the absolute orientation of the body with respect to the world frame.

The initial orientations of each body are the following (x-axis the normal vector of the sagittal plane, y-axis the normal vector of the frontal plane, z-axis the normal vector of the transverse plane) :

- For the head and the three parts of the trunk the same frame as defined in the Section 3
- for the upper arms, the forearms and the hands, the same frame as defined in the Section 3 with a rotation of $\frac{\pi}{2}$ around the y-axis.
- For the thighs and the shanks, the same frame as defined in the Section 3 with a rotation of π around the x-axis (or y-axis).
- For the feet, the same frame as defined in the Section 3 with a rotation of $-\frac{\pi}{2}$ around the x-axis.

Thereafter, the frames used to define the body position and orientation have been chosen as hypothesis:

- frame 4 for the head, at the center of the head;
- frame 3 for the upper part of the trunk, at the top of the upper part of the trunk;
- frame 2 for the middle and the lower part of the trunk, at the top of the middle part of the trunk;
- frame 8 for the right upper arm, at the elbow (top of the body);
- frame 7 for the left upper arm, at the elbow (top of the body);
- frame 10 for the right forearm, at the wrist (top of the body);
- frame 9 for the left forearm, at the wrist (top of the body);
- frame 12 for the right hand, at the fingers tips (top of the body);
- frame 11 for the left hand, at the fingers tips (top of the body);
- frame 16 for the right thigh, at the knee (top of the body);
- frame 15 for the left thigh, at the knee (top of the body);
- frame 18 for the right shank, at the ankle (top of the body);
- frame 17 for the left shank, at the ankle (top of the body);
- frame 20 for the right foot, at the center of the body;
- frame 19 for the left foot, at the center of the body;

Using this configuration, it is possible to plot the model (and his motion) using the function **ellipsoid** of Matlab (plotting ellipsoid of revolution) and the created function **elliptical** (plotting elliptical solid). A result is shown in the Figure 15.

With this animation, it can be observed that some motions have missing or corrupted data. This missing values mainly appears for quick movement, where it seems that the cameras lose the positions of some markers and is not able to correctly reconstruct the body during a certain amount of time. An example of this effect is shown in the Figure 16, it can be observed that the body is completely dislocated. Thus, computation of these motions can lead to big errors for efforts computation with the Newton-Euler algorithm as it will be shown.

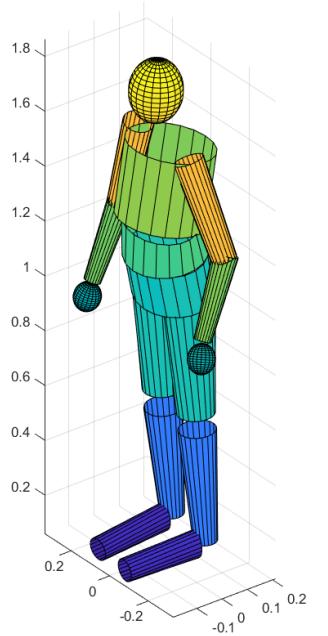


Figure 15: Created visualization

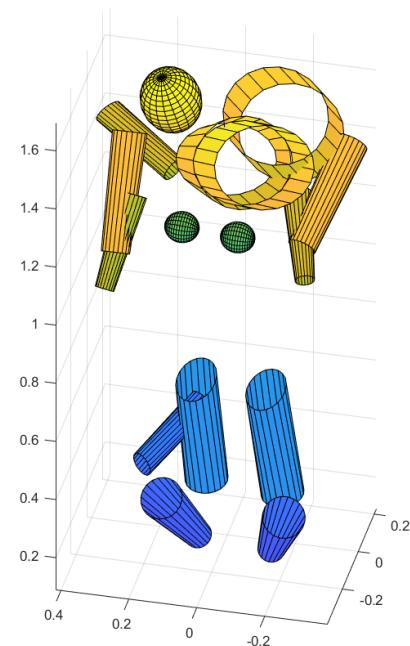


Figure 16: Fail to construct the biomechanical model for the medium jump at the 179th sample

6. Computation of the velocities and the accelerations

To compute the Newton-Euler algorithm that will be presented in the Section 7, the knowledge of the linear and angular position, velocity, and acceleration are needed.

6.1 Angular velocity and acceleration

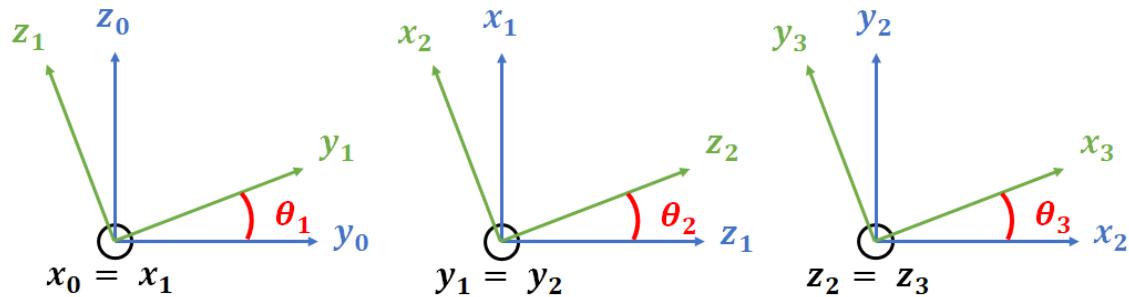


Figure 17: The three Euler rotations

The angular representation given by the data is the Newton-Euler angle of sequence "XYZ" which correspond to three successive rotations (θ_1 , θ_2 and θ_3) as shown in the Figure 17. In this picture, the world is thus the Frame 0 while the orientation of the body corresponds to the Frame 3. By noting $R_x(\theta)$, $R_y(\theta)$ and $R_z(\theta)$ the rotation matrices around the x, y and z axes, the rotation matrix from the body to the world is thus:

$${}^0R_i = R_x(\theta_1).R_y(\theta_2).R_z(\theta_3) \quad (66)$$

The angular velocity vector of the body is then determined by:

$$\begin{aligned} \omega_i &= \dot{\theta}_1.\vec{x}_0 + \dot{\theta}_2.\vec{y}_1 + \dot{\theta}_3.\vec{z}_2 \\ &= \dot{\theta}_1.\vec{x}_0 + \dot{\theta}_2.R_x(\theta_1).\vec{y}_0 + \dot{\theta}_3.R_x(\theta_1).R_y(\theta_2).\vec{z}_0 \end{aligned} \quad (67)$$

Therefore, to find the angular velocity vector of the body, the time derivative. There are three ways to compute the derivative with first order finite differences for a time step dt and a sampled signal X:

- Forward:

$$\dot{X}_k = \frac{X_{k+1} - X_k}{dt} \quad (68)$$

- Backward:

$$\dot{X}_k = \frac{X_k - X_{k-1}}{dt} \quad (69)$$

- Central:

$$\dot{X}_k = \frac{X_{k+1} - X_{k-1}}{2.dt} \quad (70)$$

The chosen method consists in using the central finite differences to compute the time derivative of the Euler angles, except for the first sample that is computed with the forward method and the last sample with the backward sample (to keep the same number of samples with the angle values).

Then the angular acceleration vector is deduced by directly applying the same method on the computed angular velocities vector.

By applying this method on the Euler angle, one will find some outliers in the velocity and acceleration values. Indeed, the Euler angle representation is a singular representation where the angles are defined between $-\pi$ and π . Thus, there are some jumps from $-\pi$ to π (or inversely) in the evolution of the angles leading to spikes in the velocity and acceleration as shown in the Figure 18. For the points before and after the gap, the value 2π must be added or subtracted (depending on the direction of the jump) to the central finite difference:

$$\dot{X}_k = \frac{X_{k+1} - X_{k-1} \pm 2\pi}{2.dt} \quad (71)$$

However, it can also happen that there two jumps happened successively letting only one value between two jumps as shown in the Figure 19. Then, for this value between the two jumps, the usual central finite difference can (and must) be apply. Finally the Figure 20 shows how this method is efficient to get rid of the outliers to recover the good signal. Thereafter, the angular acceleration vector can be computed from the computed angular velocity vector without problem. This method is used in the function `[Omega,Omega_d]=time_diff_angle(Euler_angle,t)`

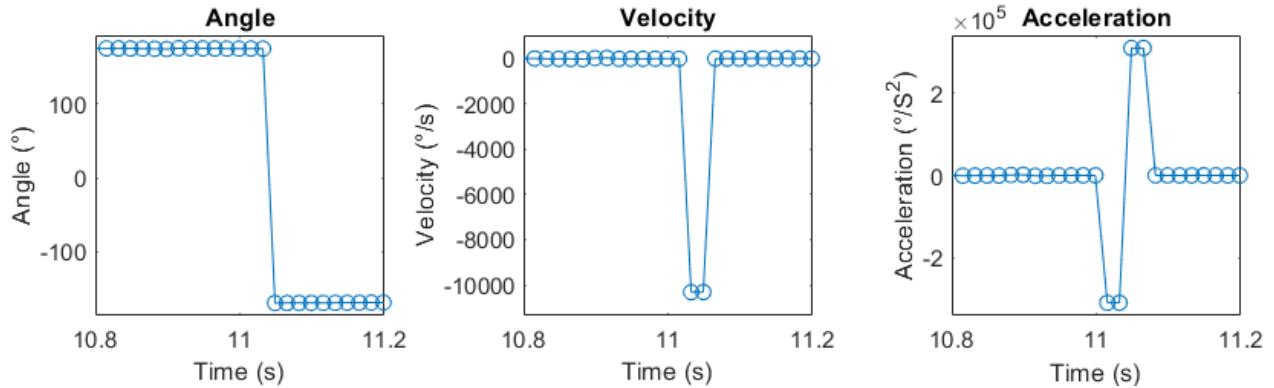


Figure 18: Angle singularities giving outliers velocity and acceleration values.

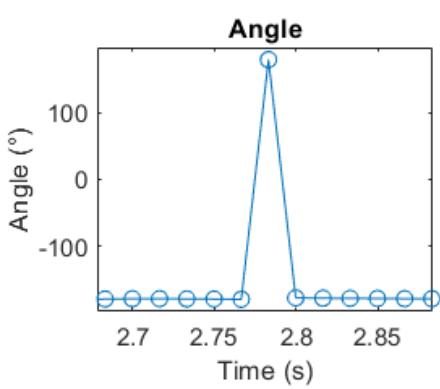


Figure 19: Value between two jumps

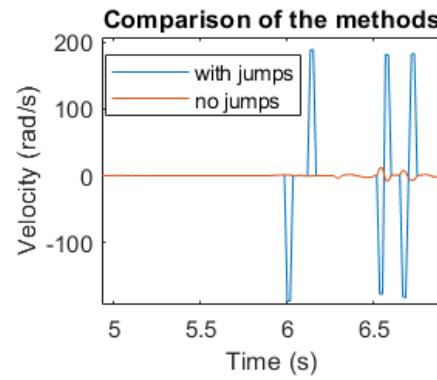


Figure 20: Comparisons of the methods with and without taking into account the jumps

6.2 Linear velocity and acceleration

The desired point to compute the linear velocity and acceleration of a body is the center of mass. Thus, a Matlab function $t=CoM_pos_orientation(segment,q)$ giving the position of the center of mass (in meters) and the absolute rotation of the body in the form $\theta\vec{u}$ (in radiant) from the frame data and the body geometry.

Then to compute the velocity and acceleration of the center of mass, the velocity and acceleration of the frames points have first been computed using the central finite differences as described in the previous section (using the $[qd,qdd]=time_diff(q,t)$ matlab function created). Thereafter the linear velocities and acceleration of the center of mass (\mathbf{v}_G and $\dot{\mathbf{v}}_G$) are computed from the frames ones (\mathbf{v}_F and $\dot{\mathbf{v}}_F$), the distance \overrightarrow{FG} from the frame to the center of mass and the angular velocities and acceleration (ω and $\dot{\omega}$) by:

$$\begin{aligned}\mathbf{v}_G &= \mathbf{v}_F + \omega \times \overrightarrow{FG} \\ \dot{\mathbf{v}}_G &= \dot{\mathbf{v}}_F + \dot{\omega} \times \overrightarrow{FG} + \omega \times (\omega \times \overrightarrow{FG})\end{aligned}\tag{72}$$

Finally, all these computations of linear and angular positions, velocities and acceleration of a body at its center of mass are computed using the function $[pos,vel,acc]=kinematic(segment,t,q)$ with inputs the segment structure described in the Section 5.1, the time vector and the frame's data.

6.3 Filtering

However, the data provided also contains some shifting, i.e. during a certain time lapse, some angles or positions are shifted of a certain angle or distance when it is compared to the data before and after. This shifting problem is illustrated in the Figure 21, for the left one it is indicated where the shifting values are and where they should be.

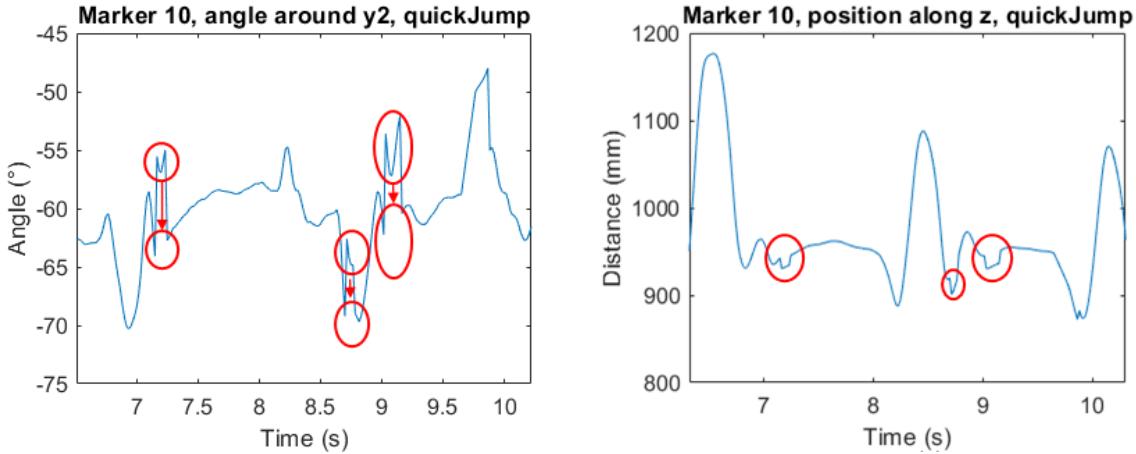


Figure 21: Examples of shifting values in the frame data

These shifting values can come from the loose of the position of some frames associated to a body on the cameras, making the analysis of the body position and orientation less efficient.

Thus, when these positions and orientations are derived, it leads to spikes velocities (and acceleration) similar at those due to the transition from $-pi$ to pi seen previously. To remove this spikes a low-pass filter wouldn't be appropriate while the high value of the spikes will still affect the around value in their mean. Thus, a median filter has been chosen. The effect of that kind of filter is that in the signal, it will look at a certain window of values around the one that will be filtered (e.g. [23, 24, 112, 26, 27] where 112 is a spike to remove), order these values (e.g [23, 24, 26, 27, 112]) and then take the median value (e.g. 26).

Through an empiric manner, a 11th-order median filter has been chosen, it is used ten times on the signal (to be sure that all the pikes are removed) then at the end a second-order Butterworth filter is used once (cut-off frequency of 20Hz, which is the order of magnitude of human movement). The purpose being to remove the spikes without affecting the rest of the signal. Also, before filtering the signals, it is verified that there is no infinite of NaN (not a number) values, otherwise they are set to zero.

This filter is implemented in `signal=signal_filter(signal,dt)` and is only applied on the velocities after their computation and before computing the accelerations.

7. The Newton Euler algorithm

The Newton Euler algorithm is a method that allows to recursively compute the torques of a robot for a given motion. This method is usually done in two way (on the following, the antecedent link is called the parent link and a successive one is called the child link):

1. A forward computation from the base of the robot to the extremities to determine the position, angle, velocities and acceleration of every part of the robot (these quantities for one body depending on parent one).

2. A backward computation from the extremities to the base to determine the efforts at every joint (the forces and torques of one body depending on the child ones).

This method can be applied for biomechanical model, either by defining a body fixed to the world (like the foot in contact with the ground) or by modeling a 6-dof (degrees of freedom) joint with one of the bodies (like the pelvis for example).

Here the first step of the Newton Euler algorithm can be avoided while the absolute positions and orientations of each body is already given.

7.1 Newton Euler algorithm for one body

The backward formulation of the Newton-Euler algorithm consists in applying the fundamental principle of the dynamic on a body, where the forces applied on it are the efforts of the previous body, the efforts of the child bodies, and the external efforts like the gravity forces or contact forces. Thus, one just needs to compute the dynamic forces and torques to apply the Newton-Euler algorithm as presented in pendulum section:

$$\begin{aligned}\Sigma \mathbf{f}_i &= m_i \dot{\mathbf{v}}_i + \dot{\omega}_i \times \mathbf{m s}_i + \omega_i \times (\omega_i \times \mathbf{m s}_i) \\ \Sigma \mathbf{m}_i &= \mathbf{I}_{O_i} \dot{\omega}_i + \mathbf{m s}_i \times \dot{\mathbf{v}}_i + \omega_i \times (\mathbf{I}_{O_i} \omega_i)\end{aligned}\quad (73)$$

Where \mathbf{v}_i and $\dot{\mathbf{v}}_i$ are the linear velocity of the body expressed at the O_i of the body i (that can be defined as the joint with the previous or the center of mass), ω_i , $\dot{\omega}_i$ are the angular velocities and acceleration of the body, m_i is the mass of the body, $\mathbf{m s}_i$ is the distance vector from the point O_i to the center of mass of the body times the mass of the body and \mathbf{I}_{O_i} is the inertia matrix of the body expressed at the point O_i .

Thus, a function $[F_i, T_i, E_c, E_p] = NE_one_body(F_j, T_j, segment, q, qd, qdd, Joint)$ has been created to compute these equations (in the folder **Newton-Euler**).

The inputs of this function are:

- F_j the sum of the forces in the joints of the child bodies,
- T_j the sum of the torques in the joints of the child bodies expressed at the point O_i where the computation is done.
- $segment$ the body structure containing the parameter of the body as defined in the Section 5.1,
- q a vector containing the position and orientation of the center of mass of the body in the world frame.
- qd a vector containing the linear and angular velocities of the center of mass of the body in the world frame.
- qdd a vector containing the linear and angular acceleration of the center of mass of the body in the world frame.
- $Joint$ the point position where the computation is done (O_i) in the world frame.

The outputs of this function are:

- F_i the force at the joint (point O_i) of the body on its parent in the world frame,
- T_i the torque at the joint (point O_i) of the body on its parent in the world frame,
- E_c the kinetic energy of the body,
- E_p the potential energy of the body.

With the input this function compute directly the distance vector \vec{S}_i from the joint to the center of mass and 0R_i the rotation matrix from the body frame to the world one. Then the linear velocity and acceleration of the point O_i are computed from the one at the center of mass (point G) by:

$$\begin{aligned}\mathbf{v}_i &= \mathbf{v}_G + {}^0\vec{S}_i \times \omega_i \\ \dot{\mathbf{v}}_i &= \dot{\mathbf{v}}_G + {}^0\vec{S}_i \times \dot{\omega}_i + \omega_i \times ({}^0\vec{S}_i \times \omega_i)\end{aligned}\quad (74)$$

(Index 0 means expressed in the world frame while index i means expressed in the body frame, if it is not indicated, it is expressed in the world frame.)

Finally to apply the equations 73, it is necessary to compute the inertia matrix expressed at the point O_i in the world frame (the one defined in the body structure is expressed at the center of mass in the body frame). First, the matrix is expressed at the point O_i using the Huygens theorem:

$${}^i I_{O_i} = {}^i I_G + m \begin{bmatrix} b^2 + c^2 & -ab & -ac \\ -ab & a^2 + c^2 & -bc \\ -ac & -bc & a^2 + b^2 \end{bmatrix} \quad (75)$$

Where a , b and c are the coordinates of the distance vector from the joint to the center of mass expressed in the body frame (${}^i \vec{S}_i = ({}^0 R_i)^T \cdot {}^0 \vec{S}_i = [a, b, c]^T$). Once this operation is done, it is only needed to express this matrix in the world frame by:

$${}^0 I_{O_i} = {}^0 R_i \cdot {}^i I_{O_i} \cdot ({}^0 R_i)^T \quad (76)$$

Thus, by noting $\vec{g} = -9.81 \cdot \vec{z}_0$ the gravity vector, it is possible to compute the desired forces by:

$$\begin{aligned} \vec{F}_i &= \vec{F}_j - m \cdot \vec{g} + \Sigma \mathbf{f}_i \\ \vec{T}_i &= \vec{T}_j - {}^0 \vec{S}_i \times m \vec{g} + \Sigma \mathbf{m}_i \end{aligned} \quad (77)$$

Finally the kinetic and potential energy are computed by:

$$\begin{aligned} E_k &= \frac{1}{2} m \cdot \mathbf{v}_G^T \cdot \mathbf{v}_G + \frac{1}{2} \omega_i^T \cdot {}^0 I_G \cdot \omega_i \\ E_p &= - m \cdot \vec{g}^T \cdot \overrightarrow{OG} \end{aligned} \quad (78)$$

7.2 Use of the Newton-Euler algorithm for one body

7.2.1 Computation of the joint torques

For motion where only one foot touches the ground, it is possible to represent the human body as successive segments linked (like a serial robot) to the ground through the foot in contact with the ground (base/root segment). Thus, the methodology here is very similar to the double pendulum problem, the forces and torques are first computed at the last link, then the parent link efforts are computed from the kinematic, the external forces and these child link efforts. Consequently, the torques of every joint can be computed from the function *NE_one_body(Fj, Tj, segment, q, qd, qdd, Joint)* with the following method:

- If the segment is the final one of a branch: the forces and torques at the point of the joint O_i are directly determined from the center of mass position of the body and of the joint, the segment parameter and the kinematic of the body using the function. The child forces and torques are set to zero while there is not a child link.
- Otherwise: the forces and torques of the child joints at the point where it is required to compute the fundamental principle of the dynamic. Thus if the body has N child bodies where there forces and torques at the point O_k , $k \in [1; N]$, these forces and torques in input of the function are:

$$\begin{aligned} F_j &= \sum_{k=1}^N F_k \\ T_j &= \sum_{k=1}^N T_k + \overrightarrow{O_j O_k} \times F_k \end{aligned} \quad (79)$$

This forces and torques get with the *NE_one_body(Fj, Tj, segment, q, qd, qdd, Joint)* function are computed in the world frame, they can be computed in the parent body of i frame (h parent to i) using the rotation matrix ${}^0 R_h^T = {}^h R_0$:

$$\begin{aligned} {}^h F_i &= {}^0 R_h^T \cdot {}^0 F_i \\ {}^h T_i &= {}^0 R_h^T \cdot {}^0 T_i \end{aligned} \quad (80)$$

However, if the joint between two bodies are spherical joints the torques in the parent body frame are directly hT_i .

It is also possible to decompose the motion as three successive revolute joints where the angles correspond to the Euler angle (${}^h\theta_1$, ${}^h\theta_2$ and ${}^h\theta_3$). This Euler angles are different from those expressed in the data file they are linked to the parent body so expressed in the parent body frame. To recover this angle it is possible to compute the rotation matrix from the body h to the body by ${}^hR_i = {}^0R_h^T \cdot {}^0R_i$ and then use the matlab function **rotm2eul** that provides the corresponding Euler angle for a rotation matrix. Therefore, the torques of each revolute joint are found by projecting the torques on each revolute joints axis:

$$\begin{aligned} T_1 &= {}^0T_i^T \cdot {}^0R_h \cdot \vec{x}_0 \\ T_2 &= {}^0T_i^T \cdot {}^0R_h \cdot R_x({}^h\theta_1) \cdot \vec{y}_0 \\ T_3 &= {}^0T_i^T \cdot {}^0R_h \cdot R_x({}^h\theta_1) \cdot R_y({}^h\theta_2) \cdot \vec{z}_0 \end{aligned} \quad (81)$$

However, this method won't be enough if the two feet touch the ground. It is also possible to use another segment as base segment (for example L_Trunk). However, to apply the Newton-Euler equations, one will have to add the forces and torques from the ground for each foot that can be get with force plates.

7.2.2 Computation of the ground reaction efforts

The ground reaction efforts can be easily computed using the *NE_one_body(Fj, Tj, segment, q, qd, qdd, Joint)* function. Indeed, the kinematic of the center of mass of each body being known, it is possible to modelize each part of the body as directly linked to the world with a 6-DoF joint at the center of mass to compute independently the forces and torques of each body on the ground, rather than propagate the force and torques from one body to the other until reaching the ground.

Thus, for each body the input of the *NE_one_body(Fj, Tj, segment, q, qd, qdd, Joint)* will be:

- $F_j = 0_{3x1}$, $T_j = 0_{3x1}$,
- segment, q, qd, qdd: the corresponding segment structure and kinematic of the center of mass,
- Joint: the position of the center of mass of the body.

Once the forces and torques of each body have been computed at their center of mass (G_k $k \in [1, 17]$), they are summed at the origin of the world frame $O_0 = [0; 0; 0]^T$ to find the round reaction efforts:

$$\begin{aligned} F_0 &= \sum_{k=1}^{17} F_k \\ T_0 &= \sum_{k=1}^{17} T_k + \overrightarrow{O_0G_k} \times F_k \end{aligned} \quad (82)$$

This data can thereafter be compared to the one from the force plate, however the positions and orientations of the reference frame of the force plate is different from the ones of the motion capture system. Thus, one must find the relative position and orientation of the two systems reference frames. This is done in the script *NE_full_body*

8. Analysis of the results

8.1 Alignment of the reference frames and first analysis

Running the codes *NE_full_body* for the **quick jump** motion as described previously will give the forces presented in the Figure 22. It is significant the the forces along the z-axis have very similar shapes but are

shifted in time. Thus, the motion capture signals must be shifted in time in order to be able to compare the forces.

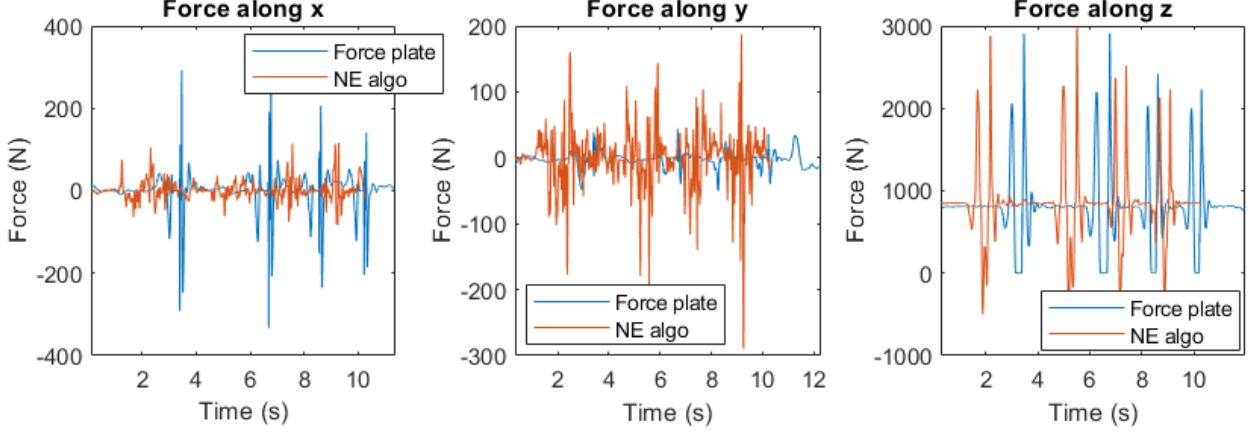


Figure 22: First try with the quick jump motion

The alignment of the time, for both signal, leads to the curves in the Figure 23. It can be seen that the forces along the z-axis are very similar and have the same order of magnitude. But while comparing the x-axis and y-axis, the forces for the first one has higher amplitudes for the force plate data but for the y-axis it is the force from the Newton-Euler computation that has higher values. Thus, it is possible that a rotation of the frame is needed in order to compare the appropriate values.

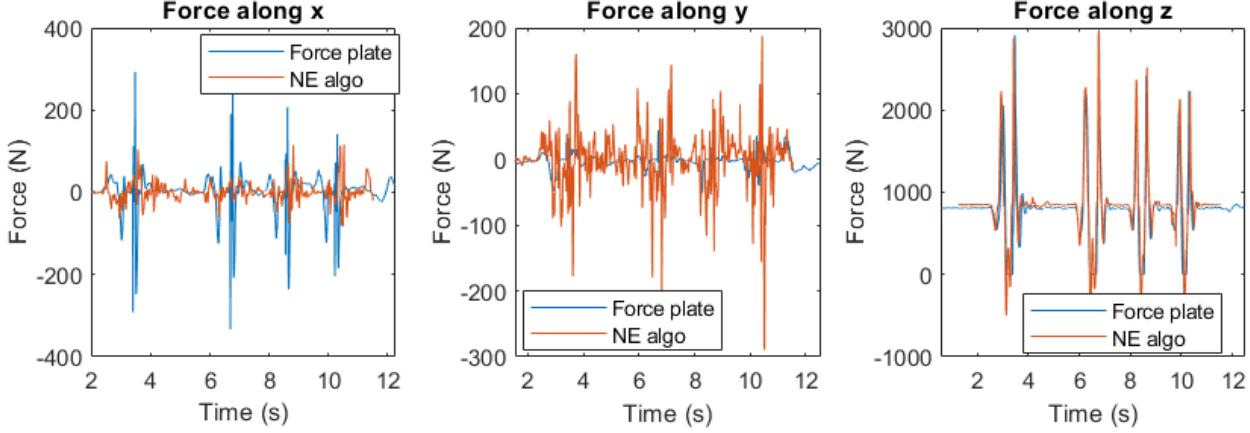


Figure 23: Time alignment

Rotating the frame of the Newton-Euler algorithm computed efforts of $\frac{\pi}{2}$ around the z-axis leads to the curves presented in the Figure 24.

First, it is observed that the forces along the z-axis are very similar except for the moments where the force from the force plate is null (corresponding to the moment where the subject is in the air for this jumping motion). At these moments, the NE (Newton-Euler) computed force seems to oscillate, that effect can be linked to the fact that the markers are not fixed on the skin (nor the skin on the bones). Indeed, during the jumping the unfixed marker can oscillate on the body, leading to oscillating bodies viewed by the motion capture system leading to oscillating acceleration and thus oscillating forces.

For the forces along the x and y axes the shapes doesn't fit as well as for the z-axis, one could see for the y-axis force a similar shape at the end and also a similar shape at the beginning through a noise (by doing the sliding average). However, for the x-axis, it can be noticed that the order of magnitudes of the spikes are really similar and that these spikes happened at the same time for both data.

Globally for the x and y axes forces the computed forces seem to have a lot of spikes (noise) that can be due to the modelling errors coupled to the undesired motion of the markers on the skin.

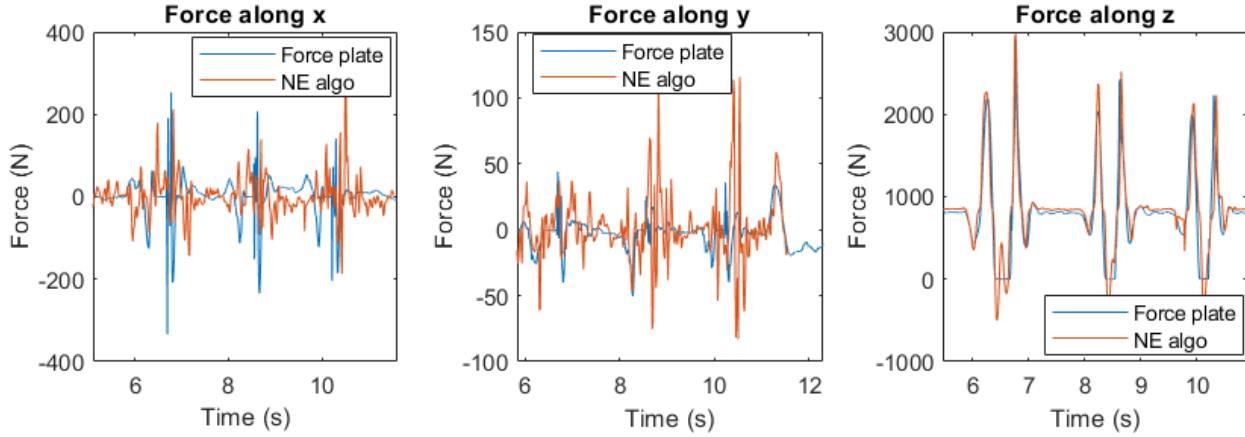


Figure 24: Frame orientation alignment

Now that the orientation of the frames are aligned, the question of the alignment of the position of the frames is needed in order to compare the torques, a point that can be used is the center of pressure (CoP). Indeed, the force plate data gives the position of the center of pressure of the body. Thus, by computing the center of pressure of the body with the Newton-Euler algorithm, it is possible to find the position of a common point in the two systems frames and thus, the relative positions of the two system reference frames. Thereafter by computing the torques at these points for the force plate data and computed torques, an analysis is possible.

To determine the center of pressure for the computed ground reaction efforts, it is supposed that the body is in equilibrium, then this point correspond to the ZMP (Zero Moment Point) where the momentums around the x and y axes are null. The position of this point ($[x_p, y_p, 0]^T$) can be found from the forces and torques at the reference frame (which is on the ground) by:

$$\begin{aligned} x_p &= -\frac{T_y}{F_z} \\ y_p &= \frac{T_x}{F_z} \end{aligned} \tag{83}$$

Where $T_y = T_0^T \cdot \vec{y}_0$, $T_x = T_0^T \cdot \vec{x}_0$ and $F_z = F_0^T \cdot \vec{z}_0$.

Then by moving all the torques at that point, the curves of the Figure 25 are obtained. One can see on these curves that the torques around the x and y axes are null as expected (except for the moments where the body is in the air for the force plate data). However, for the z-axis, the computed torques have very higher values than the force plate data and seems very noisy too as for the forces.

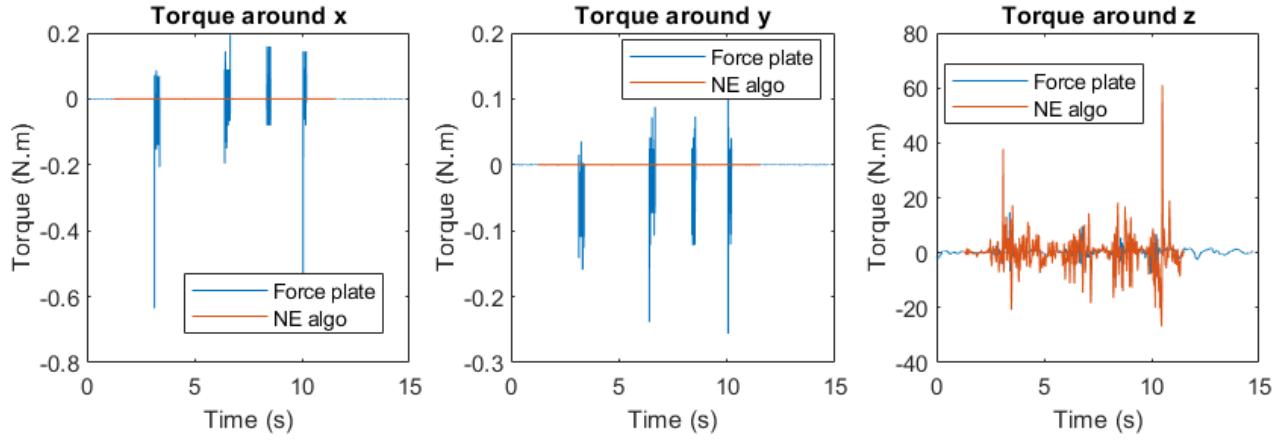


Figure 25: Frame position alignment

Question of the data sampling

Not only the time duration and initial time are different between the force plate data and the motion capture data but also the sampling frequency is higher for the force plate. Thus, for being able to print the error between the force plate data and the computed efforts, one could think to resample the force plate data. This resampling can be done with the `[signal, T]=signal_resampling(signal, T, newT)` created matlab function and an example is given in the Figure 26.

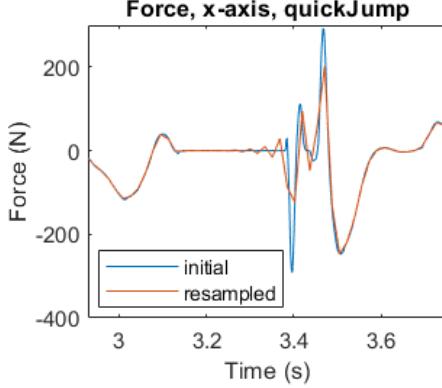


Figure 26: Example of the force plate data resampling

In this Figure, it is possible to see that the signal is greatly affected when high variations appear. Therefore, in the following work the resampling won't be done and the analysis will only be done by comparing the curves by superposing them.

8.2 Influence of the dynamics on the noise

The Figures 27, 28 and 29 show a waving arm motion for three different velocities.

First, for the slow motion (Figure 27) it can be seen that the force curves have very similar form:

The computed forces along the x-axis present the same two spikes than the force plate data (just before

13.5s) and present some oscillations before and after the spikes that are not present on the force plate data (these oscillations are also due to the model errors and the markers gliding on the skin).

The computed forces along the y-axis are very similar to the data ones between 12 and 13.1 seconds. However, after the rise, some oscillations with a big spike appear, then the fall of the force is visible but also followed with reduced oscillations before stabilizing to the zero again (with a bit of noise). These oscillations following the rise or the fall are directly linked to the movement of the markers on the skin and the model errors.

On the z-axis force curves, it can first be seen that when there is no movement, there is a little shift between the two forces, this shift is linked to the computation of the mass of the bodies that led to 86 kg and not 80 kg (the real mass of the subject) as previously observed in the section 3. It can also be seen, for this axis, that the spikes of the computed force are lower than those from the data, this result is also present for the other speed in the Figures 28 and 29. This result is in opposition with the ones from the jumping motion that has been seen in the Figure 24, an explication could be that for the jumping motion there is very high dynamic making that some acceleration errors would increase the amplitude, but nothing is proven and it is just a theory. However, an explanation of this lower amplitude can come from the filtering of the velocities that would have diminished some spike that would not be errors. Indeed, by comparing with the forces obtained without filtering in the Figure 30, it is visible that the amplitude of the z-axis forces spike corresponds to the one from the data, however the resulting forces for the x and y axes become noisier with very high spike and it is not possible to do an analysis with these data.

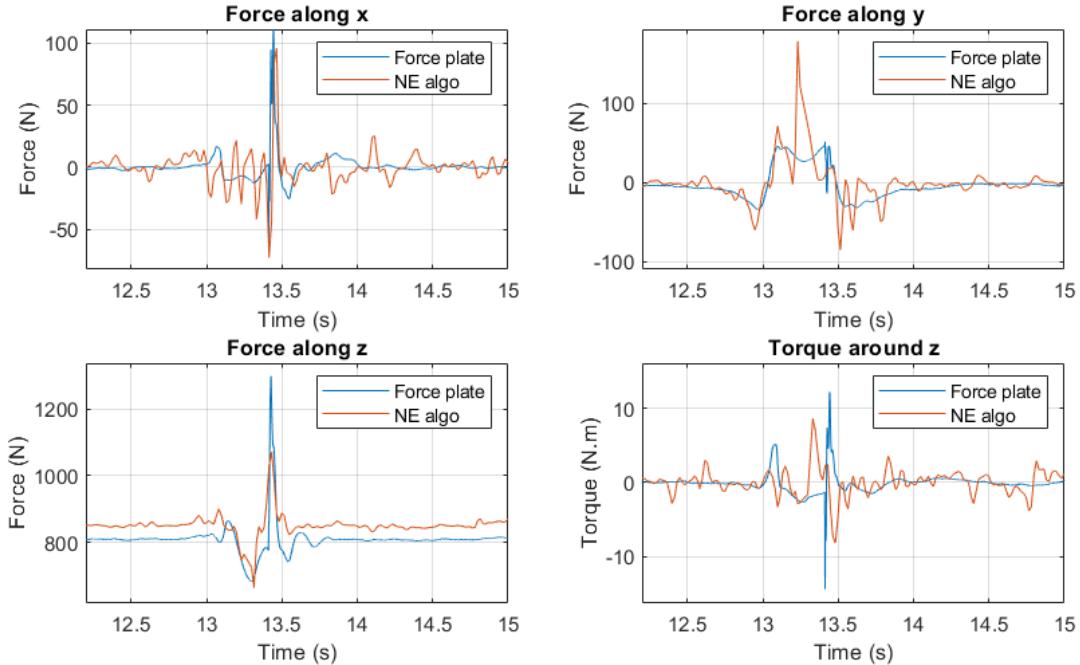


Figure 27: Ground reaction efforts for the slow arm motion

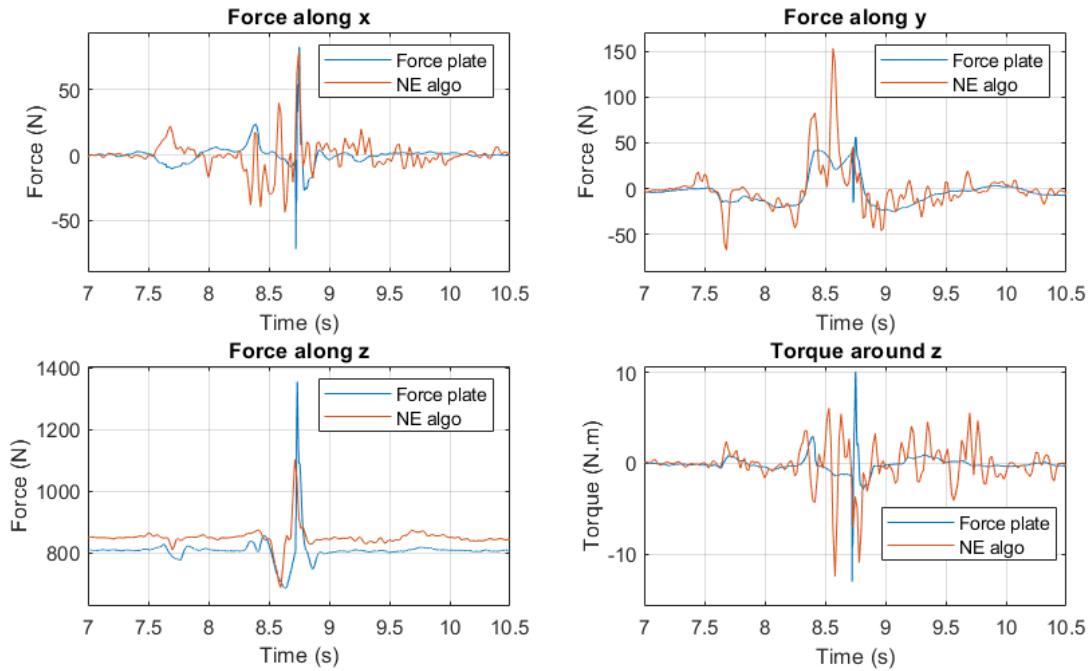


Figure 28: Ground reaction efforts for the medium arm motion

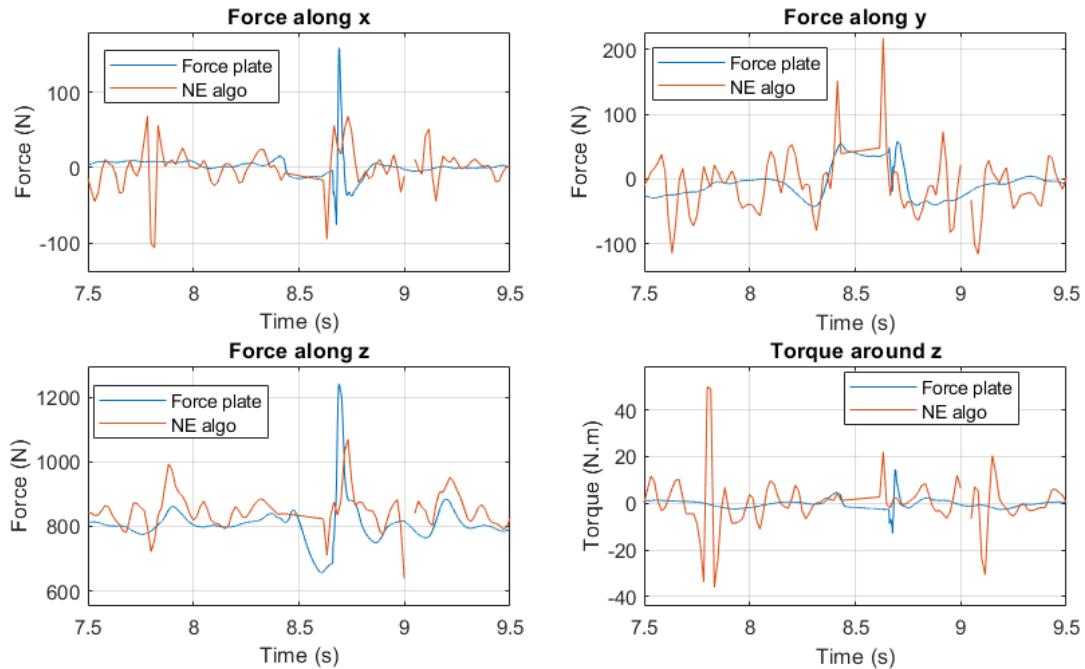


Figure 29: Ground reaction efforts for the fast arm motion

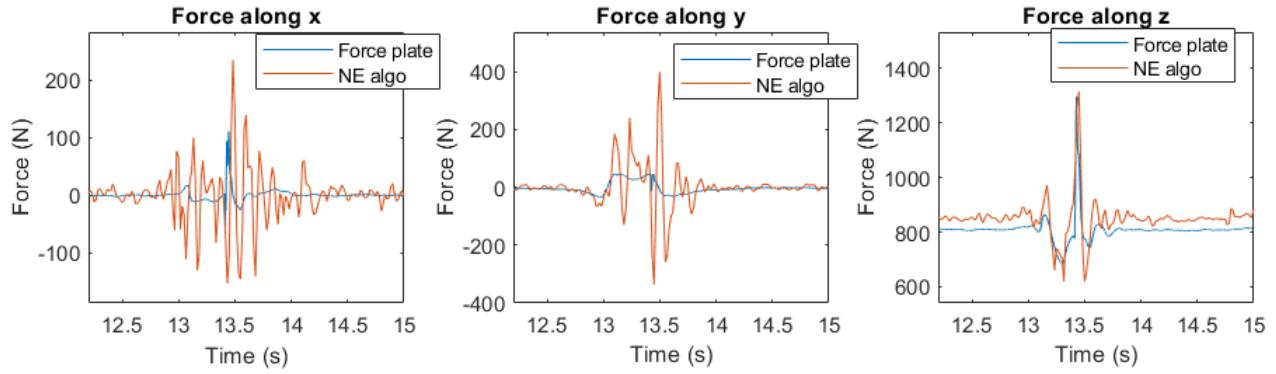


Figure 30: Ground reaction efforts for the slow arm motion with non-filtered velocities and acceleration

Finally, for the torque around the z-axis, it is more difficult to associate the shape of the computed torque to the data one, however when the full motion is observed as in the Figure 31, the shape of the force plate torque can be seen through the oscillation for the computed torque. Notably, on this Figure, it can be seen that the oscillations generally appear when there are movements. These oscillations are higher for the torques than for the forces because there are more accumulated errors (and thus accumulated noise) when the torques are expressed at the computed center of pressure. Indeed, the position of this point containing a lot of error while it is computed from the computed force and torques.

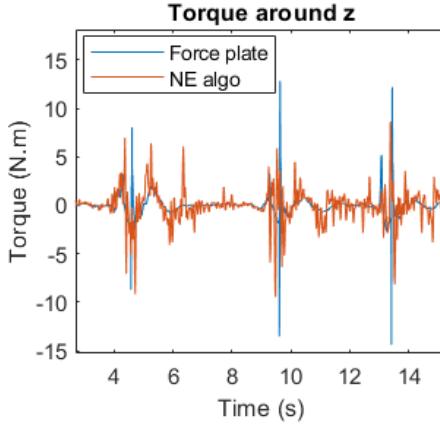


Figure 31: Ground reaction torque for the slow arm motion

Now, by comparing the result for different speed of the movement, similar results are observed for the medium and fast movement. However, the amplitude of the oscillations (the noise) is higher for the medium movement than for the slow movement and greatly higher for the fast movement (the amplitude of oscillations for the torques becomes ten times superior to the highest peak of the data from the force plate).

Therefore, it can be concluded that the speed of the motion is linked to the amplitude of the noise. Indeed, in one hand because of the model error, the computed mass and inertia don't correspond to the reality and these errors become more important when the accelerations are high. In another hand, because of the speed of the movement the marker suit can move more on the body generating additional errors.

The Appendix C presents the result of the ground efforts computation for different motions. The influence of the dynamic of the noise can also be seen on the force along y for the kick and kick-arm motions. However,

for these motions the force along x is noisier than for only arm motion, it can be due to the orientation data of the leg that can be noisier.

8.3 Computation of joint torques

The **NE_arm.m** script implements the computation presented in the Section 7.2.1 for the right arm. This script first computes the torques in the world frame for the wrist, the elbow and the shoulder, then the relative Euler angle between two bodies are computed and at the end the torques are projected on the axes of these Euler angles.

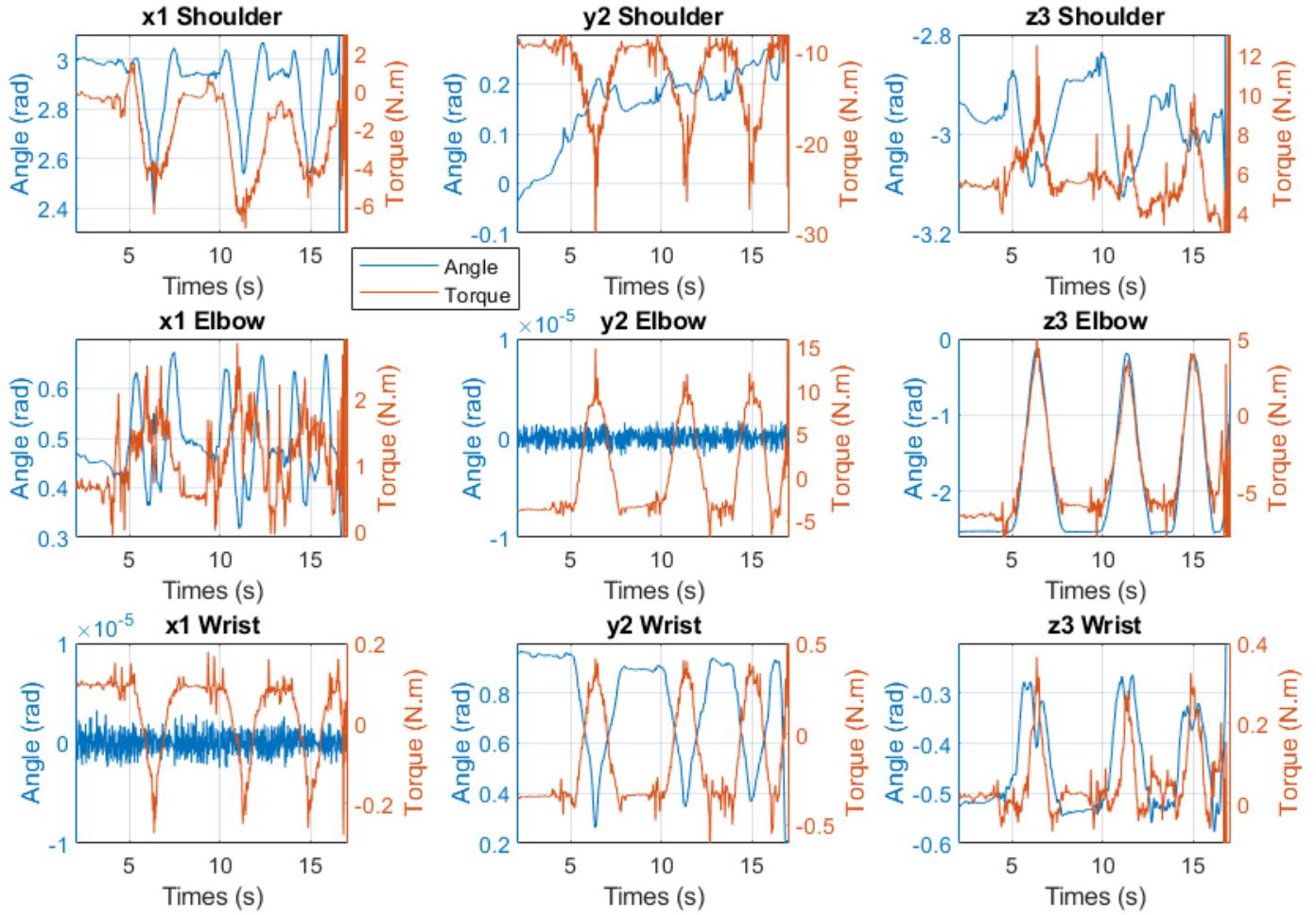


Figure 32: Angle and torques around the relative euler angle for the slow arm motion

The result of this script for the slow arm motion is presented in the Figure 32. In this Figure, the first line corresponds to the angle and torques around the three Euler angle axes for the shoulder (angle between the upper trunk and the upper arm), the second line are the ones for the elbow and the third line are the ones for

the wrist. The axes are noted x_1 , y_2 and z_3 as presented in the Figure 17.

In this Figure, it is observed that the angle around the x -axis is blocked for the wrist, this axis corresponds to the one along the forearm, therefore this result corresponds to the reality. The elbow is usually represented by a revolute joint (around the z -axis in our case as presented in the Figure 33), but a rotation exists in the forearm along the x -axis due to the radius and ulna bones, this internal rotation of the forearm is here associated to the elbow according to the curves. Thus, the rotation around the y -axis is blocked for the elbow.

This motion being slow, the torques in the arm are mostly due to the gravity force on the arm and not to the dynamic. Consequently, the evolution of the torques are really similar to the evolution of the z -axis angle of the elbow, the movement being done mostly around this axis as it can be seen through the angles amplitudes (it can also be seen in the animation). These torque shapes are well visible for all wrist torques, the y and z axes elbow torques and a bit for the shoulder torques. Indeed, because of the arm motion the gravity force moves in the bodies frame and thus the angle axis supporting the torques due to this force change.

The elbow torques have slightly different shapes (compared to the z -axis elbow angle) since this articulation accumulates the forces and the torques of all the child bodies (upper arm, forearm, hand). For the same reason, the maximum torque values are less for the wrist than for the elbow and less for the elbow than for the shoulder.

Therefore, the computation of the torques is possible and the method can be extended to almost all the body. Indeed, when the two legs are on the ground, the division of the efforts between the two legs can't be determined unless the ground reaction efforts are known for both the legs. Also, the presented curves are less noisy than the ones from the ground reaction efforts computation due to the fact that less bodies are considered in this problem (thus less accumulated noise).

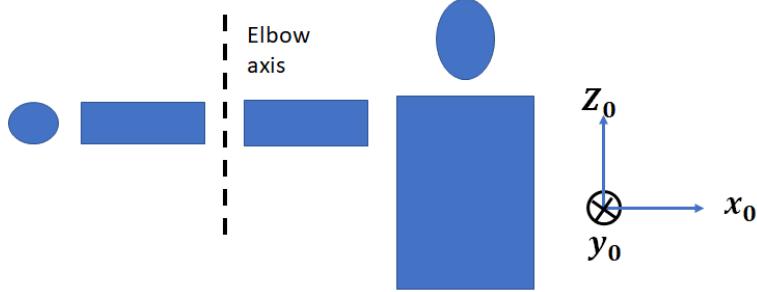


Figure 33: Initial position of the right arm (all angles null)

8.4 Study of the energy for the jumping motions

The script *NE_full_body* also allows to compute the energies of the full body during the movement, this energy is particularly interesting in the case of jumping motion. The Figures 34, 35, 36 and 37 correspond to the energy curves for the different types of jump. Also, because of the high dynamics for this type of motion, some data are lost (especially for the maximum jump and the jump with feet up). However, it can also be seen that there is less noise because the computation of the energy uses the velocities and not the acceleration like the computation of the efforts. Indeed, the acceleration is noisier than the velocity because the derivation of the noise causes a higher noise.

These figures show the kinetic, potential and mechanical (sum of the potential and kinetic) energies. It can be noticed that during the jump there is a transfer of energy between the kinetic and potential energy leading to a more or less constant mechanical energy. Indeed, there is first a peak of kinetic energy where the subject is shooting his body in the air. Then, the kinetic energy becomes null while there is a peak of potential energy when the body is at the maximum height of its jump before going down. Finally, there is a peak of kinetic energy just before the body touches the ground.

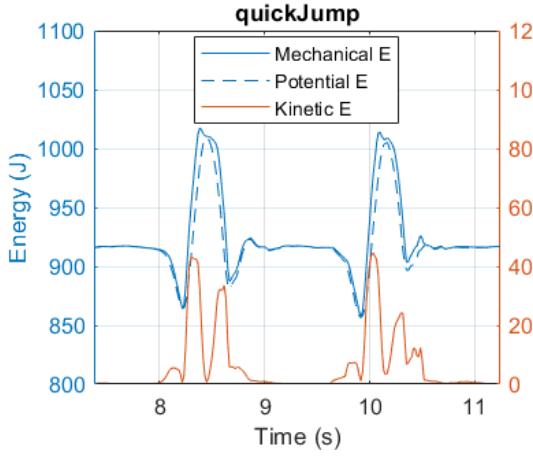


Figure 34: Energies for quick jump

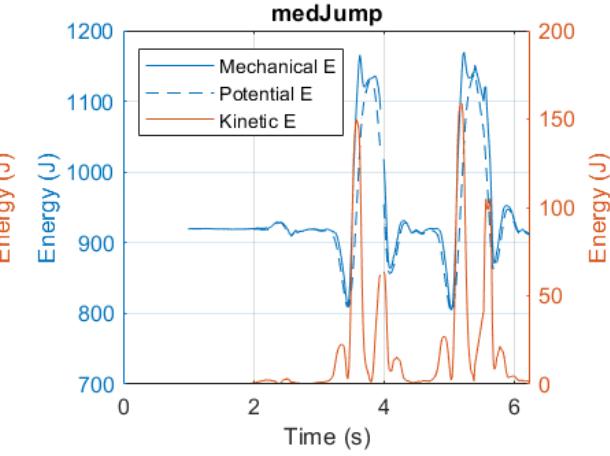


Figure 35: Energies for medium jump

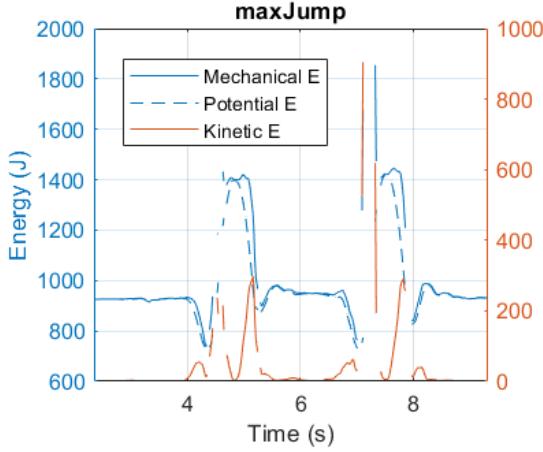


Figure 36: Energies for maximum jump

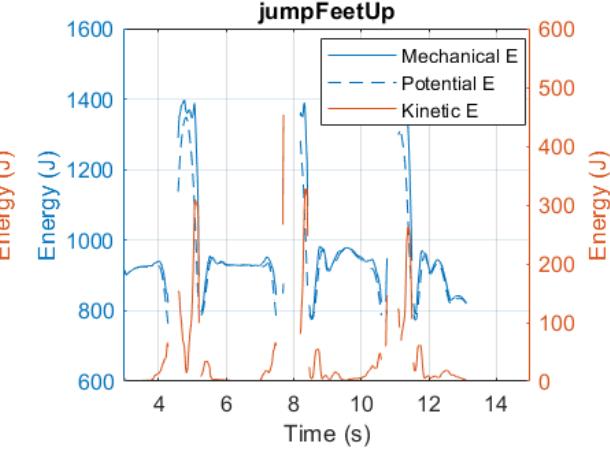


Figure 37: Energies for jump with feet up

On these curves, the potential energy diminished before the jump, this is due to the bending of the knee of the subject before the jump lowers his center of gravity. This lowering is also present (with a lower amplitude) when he arrives on the ground to damp the shock before standing tall again. Because the subject bends more his knees before jumping than after jumping, it can also be noticed that the kinetic energy peak is higher at the beginning of the jump than at the end.

Thus, by comparing the mechanical energy when the subject is bending the knee before the jump and the mechanical energy when the subject is in the air, it is possible to estimate the energy that the subject uses to shoot is body in the air:

- For the quick jump, this energy is about 150 J ($\sim 1020 - 870$ J),
- For the medium jump, this energy is about 310 J ($\sim 1120 - 810$ J),
- For the maximum jump, this energy is about 620 J ($\sim 1400 - 780$ J),
- For the jump with feet up this energy, is about 600 J ($\sim 1380 - 780$ J).

It is also possible to estimate the height that the center of mass of the body reaches from its initial position by comparing the mechanical energy when the subject is standing without moving and the mechanical energy when the subject is in the air. Indeed, this variation of energy can be seen as a work of the gravity force:

$\Delta E_m = (m.g).\Delta h$. With a standing mechanical energy of 920 J, a mass of 86.6 kg and $g = 9.81 m.s^{-2}$, the height variation of the center of mass is about:

- $\Delta h = 12\text{cm}$ for the quick jump,
- $\Delta h = 24\text{cm}$ for the medium jump,
- $\Delta h = 57\text{cm}$ for the maximum jump,
- $\Delta h = 54\text{cm}$ for the jump with feet up,

However, this variation of the height of the center of mass shouldn't be confound with the height that the subject can reach with his head. Indeed, for the maximum jump and jump with feet up motions, the subject is raising his arms or his legs raising at the same time the position of the center of mass in the body. To determine the height that the head is reaching, this method can be used with only the energy of head and not the full body.

9. Conclusion

This report has shown how to modelize a human body before applying this model on a motion capture system. Thereafter, from this biomechanical model and the motion data, it is possible to first determine the kinematic of the bodies composing the model and the efforts between the bodies and the environment, thus, the human body articulation and the ground reaction efforts.

However, it is important to highlight that the measurement method of the Hanavan parameter must be rigorous to have the best fitting model to avoid dynamic errors. For the same reason, the subject should also wear the markers on the skin to reduce as much as possible the movement of the markers with respect to the bones.

Despite these errors, it has been seen that the proposed method to compute the efforts can still be used to estimate the ground reaction forces for slow motions or in a moderate way the medium speed motion. At the opposite, it is hard to know if the computed torques are acceptable while the only point of comparison for the computation of the torques (the center of pressure) also contains errors (when it is computed) leading to very noisy torques when they are computed at this point. Though, the computation of joint torques provides also good results, and it is possible to determine the torques for the whole body for slow motion with very few noises.

However energetic study can be performed even for high-speed motion because they depend and the positions and velocities data that are less noisy than the accelerations data.

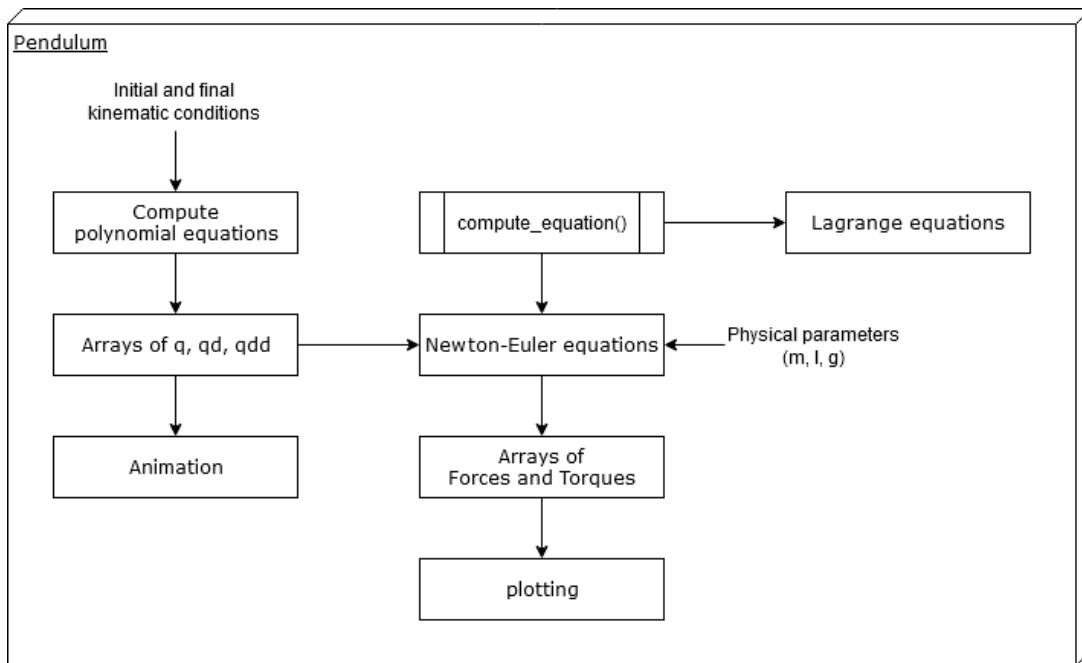
A. Folder organization

The work is separated between 6 folders:

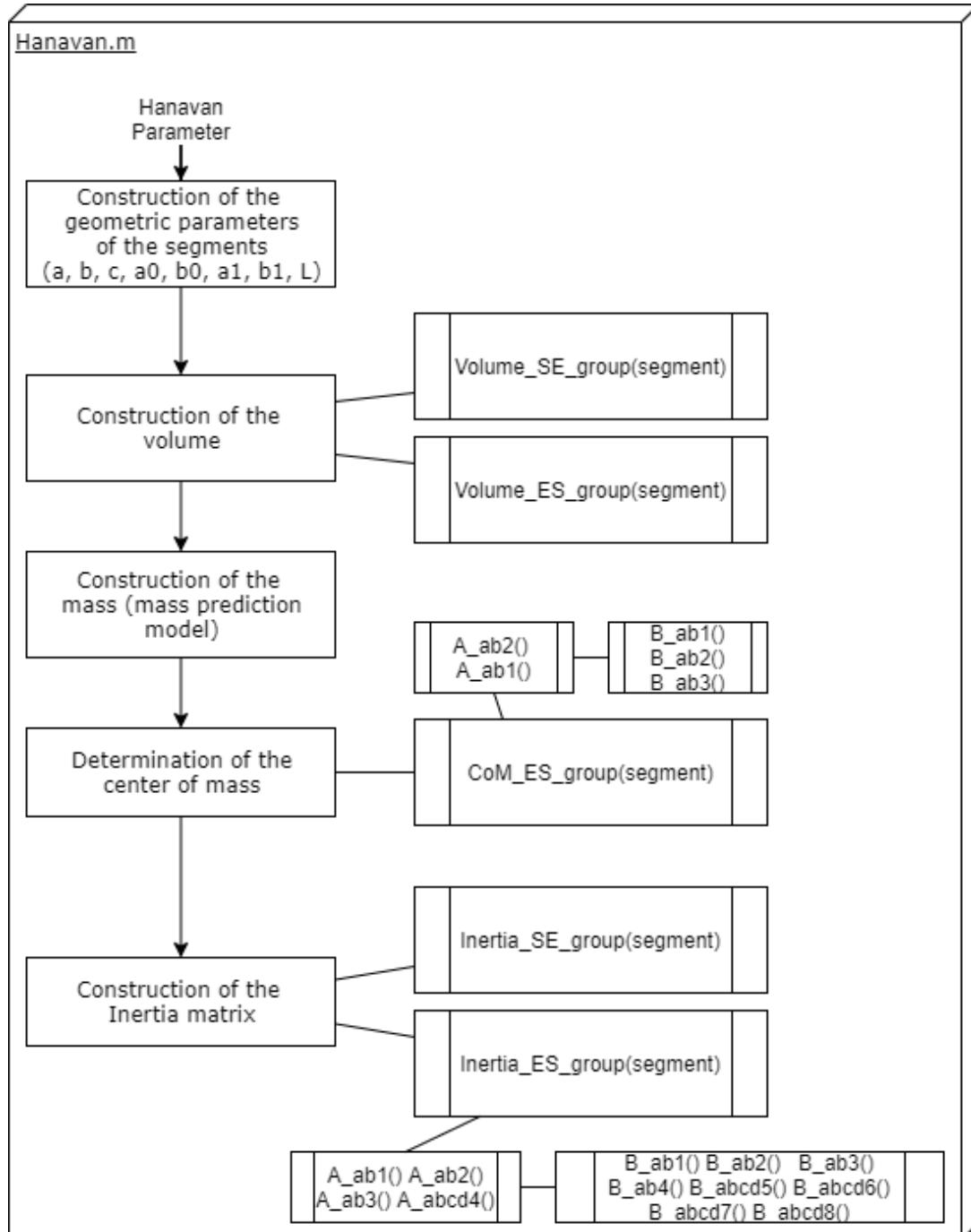
- The **Pendulum** folder containing the *simple_pendulum.m* and the *double_pendulum.m* file with the implementation presented in the Section 2,
- The **Hanavan model** folder containing the *Hanavan.m* script with all the construction of the model from the Hanavan parameter,
- The **Reading files** folder with the function *read_csv.m* and *read_drf.m* permitting to get the data from the force plate and the motion capture, and the **ground** and **motion** folders containing the force plate data and motion data for the different movements under Matlab form,
- The **Animation** folder with the *Animation.m* script permitting to have a visualization of the motion,
- The **Newton-Euler** folder with the *NE_fullbody.m* and *NE_arm.m* script applying the Newton-Euler algorithm to get the ground reaction efforts or the arm torques.
- The **Function** folders containing all the function used by the scripts.

B. Workflow

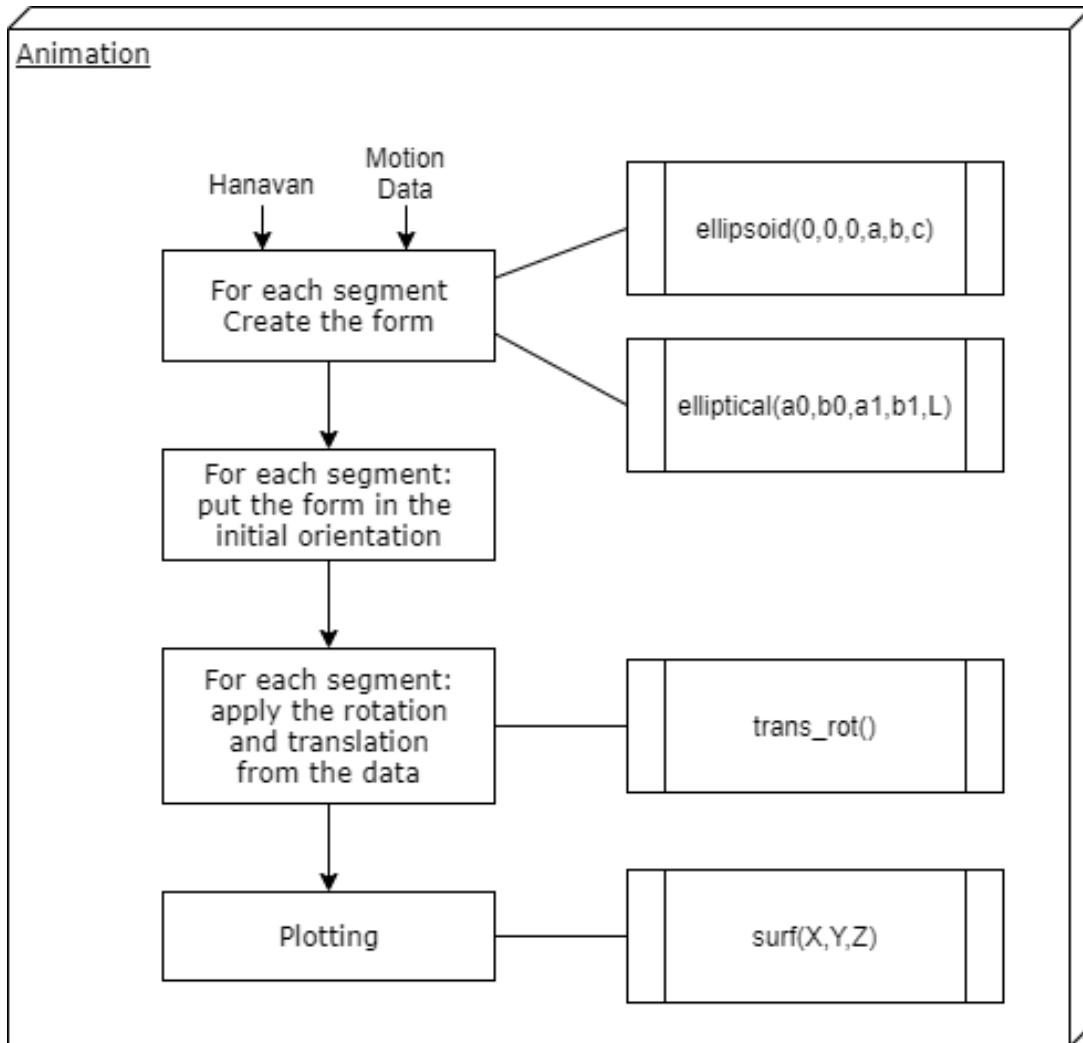
B.1 Pendulum



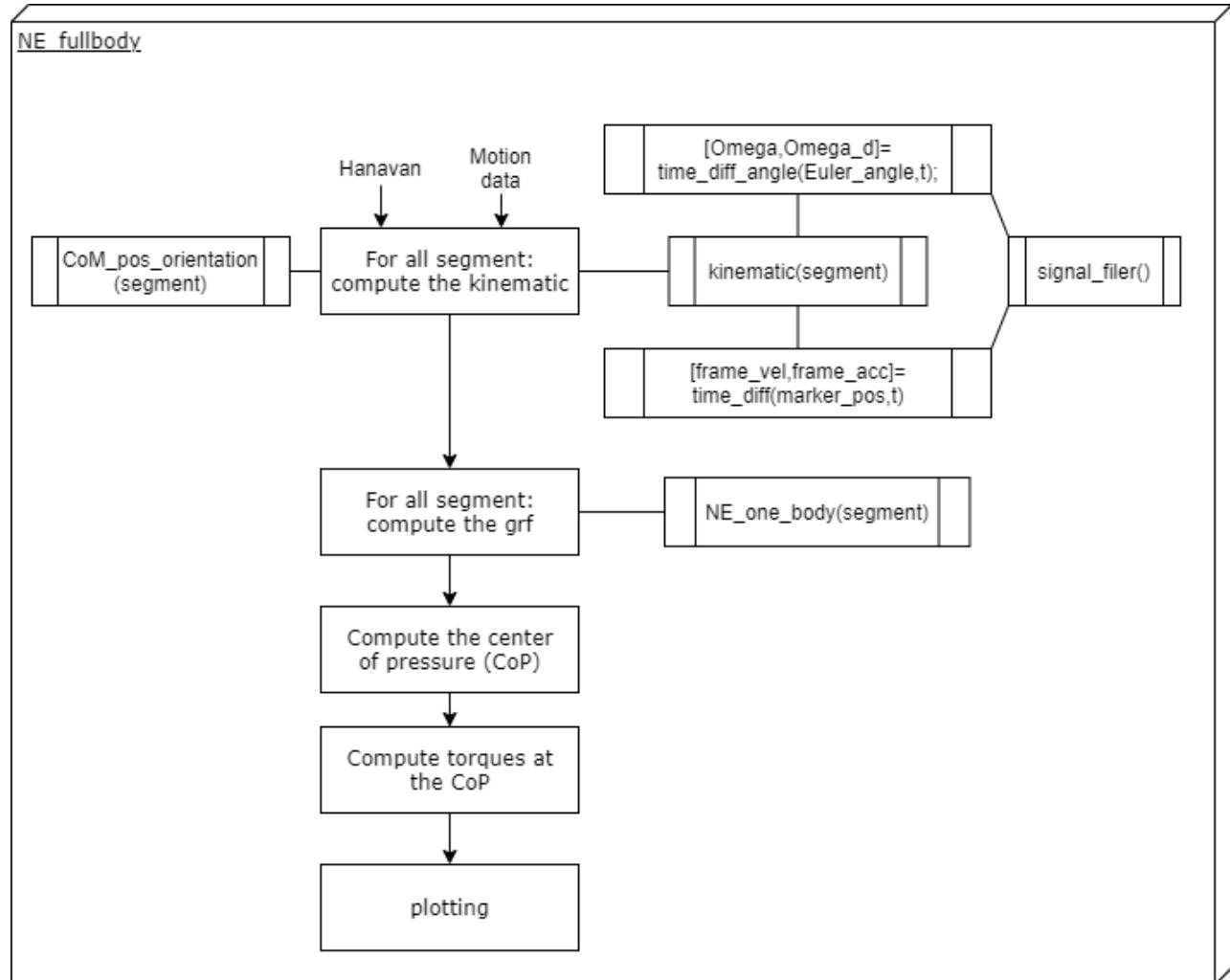
B.2 Hanavan.m



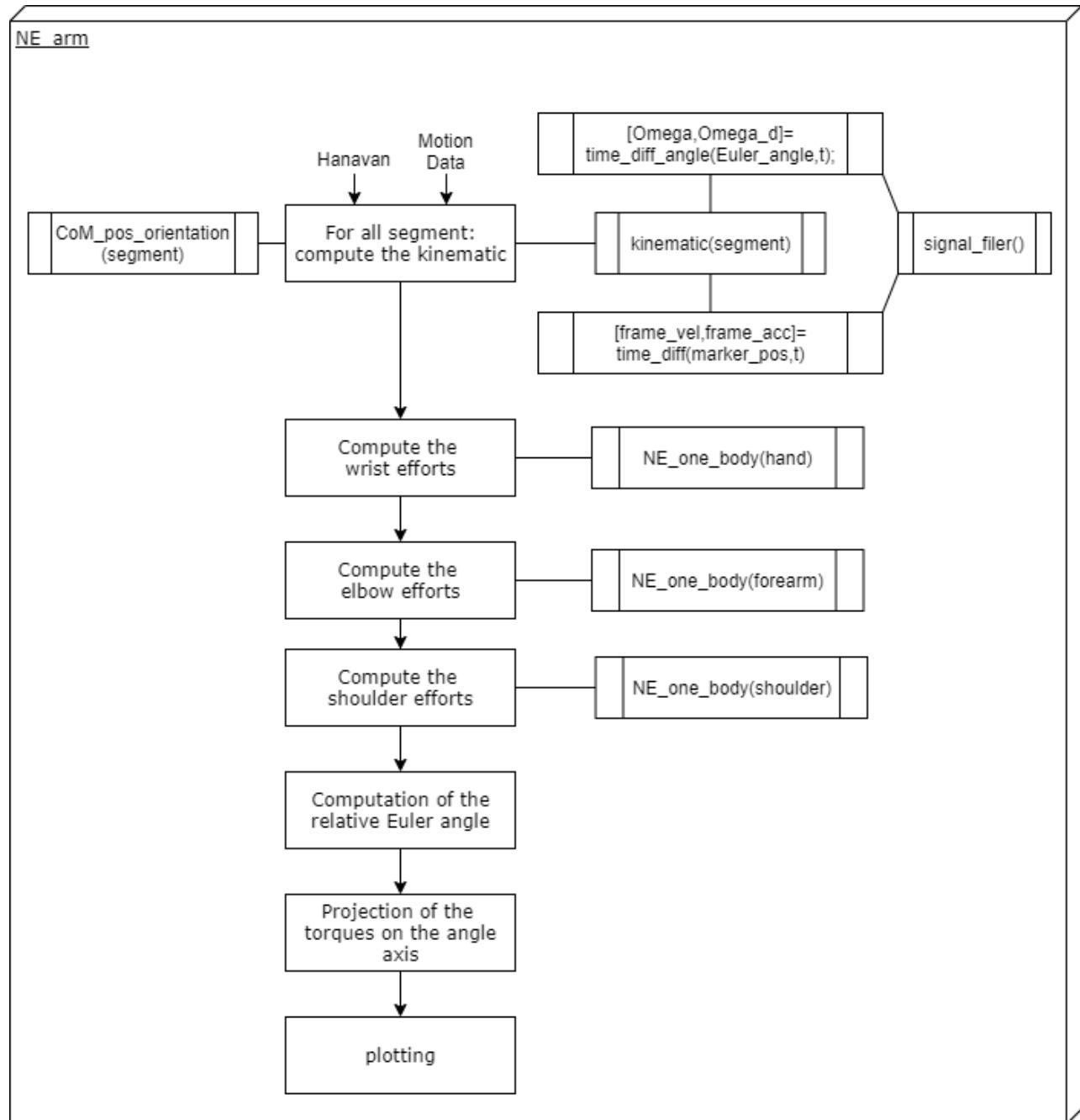
B.3 Animation.m



B.4 Ne_fullbody.m



B.5 Ne_arm.m



C. Ground reaction forces for some motion

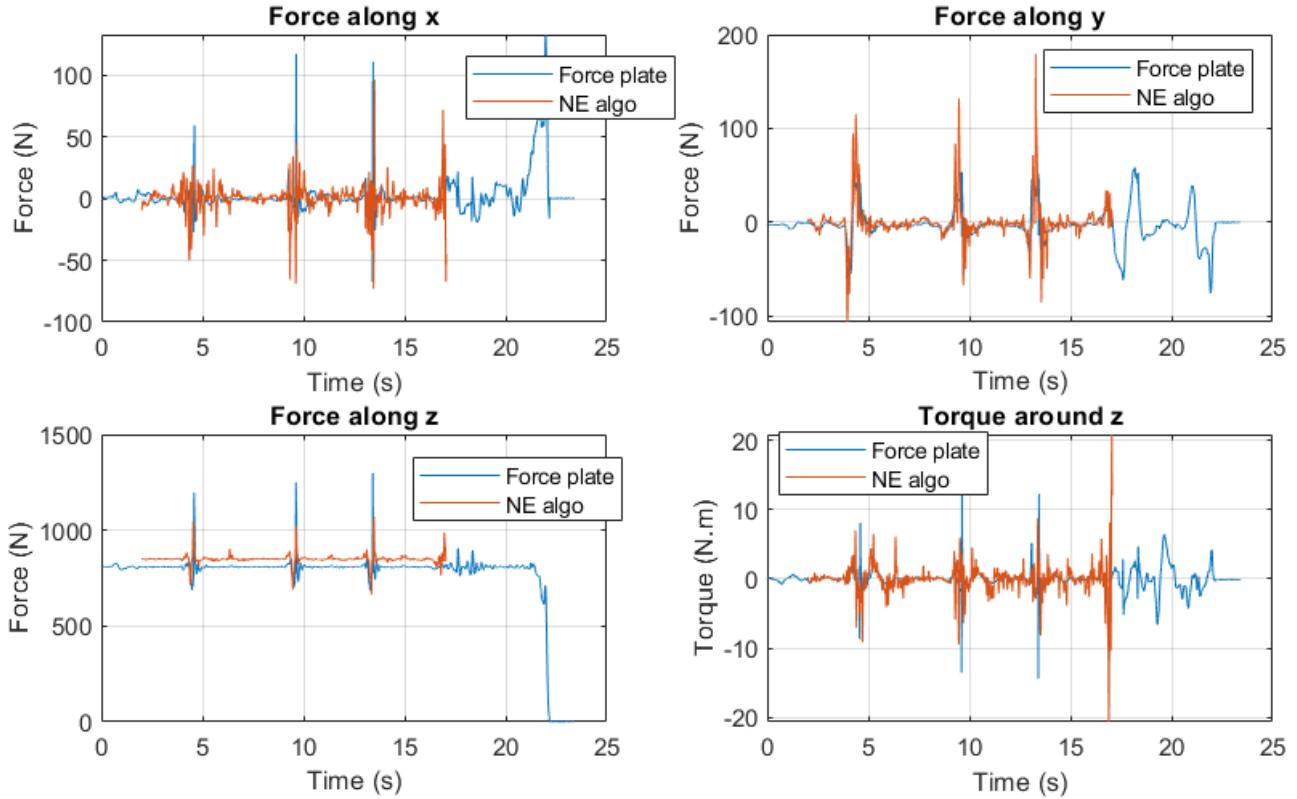


Figure 38: Slow arm

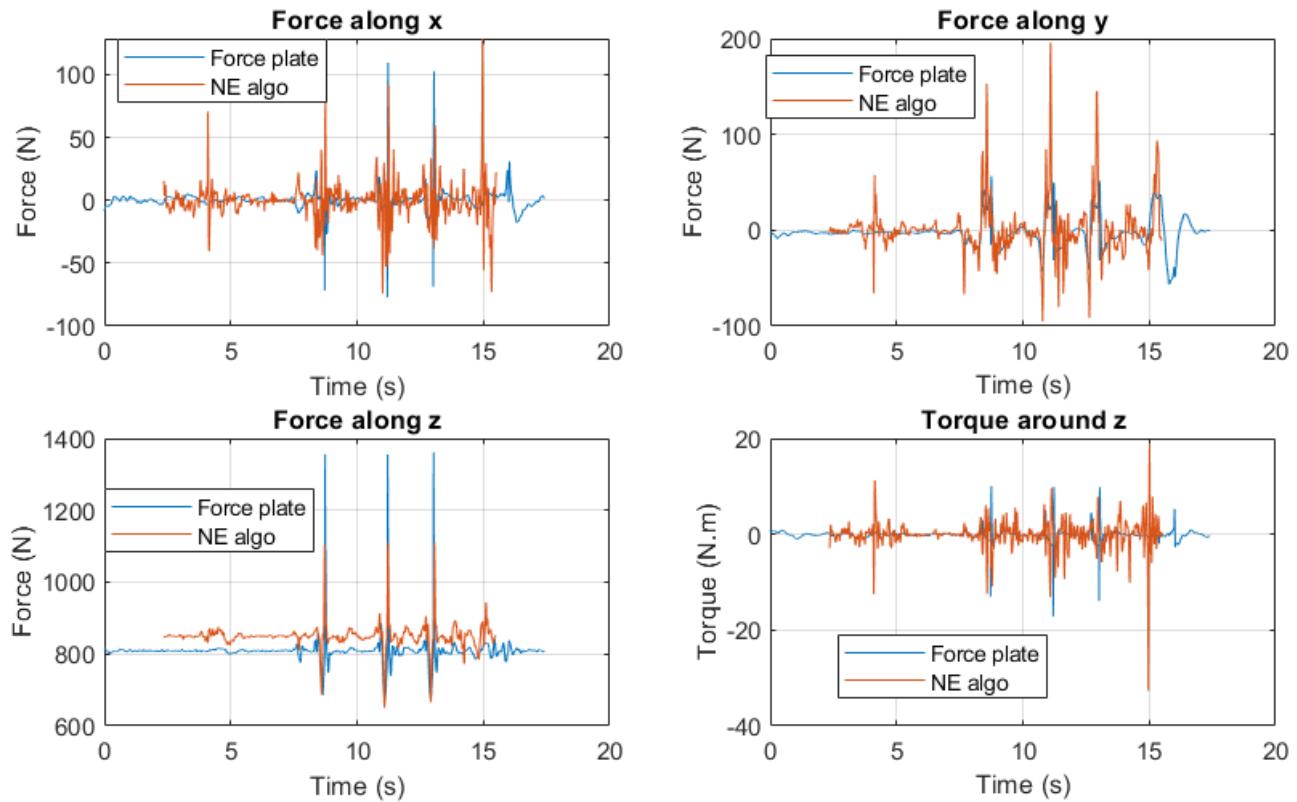


Figure 39: Medium arm

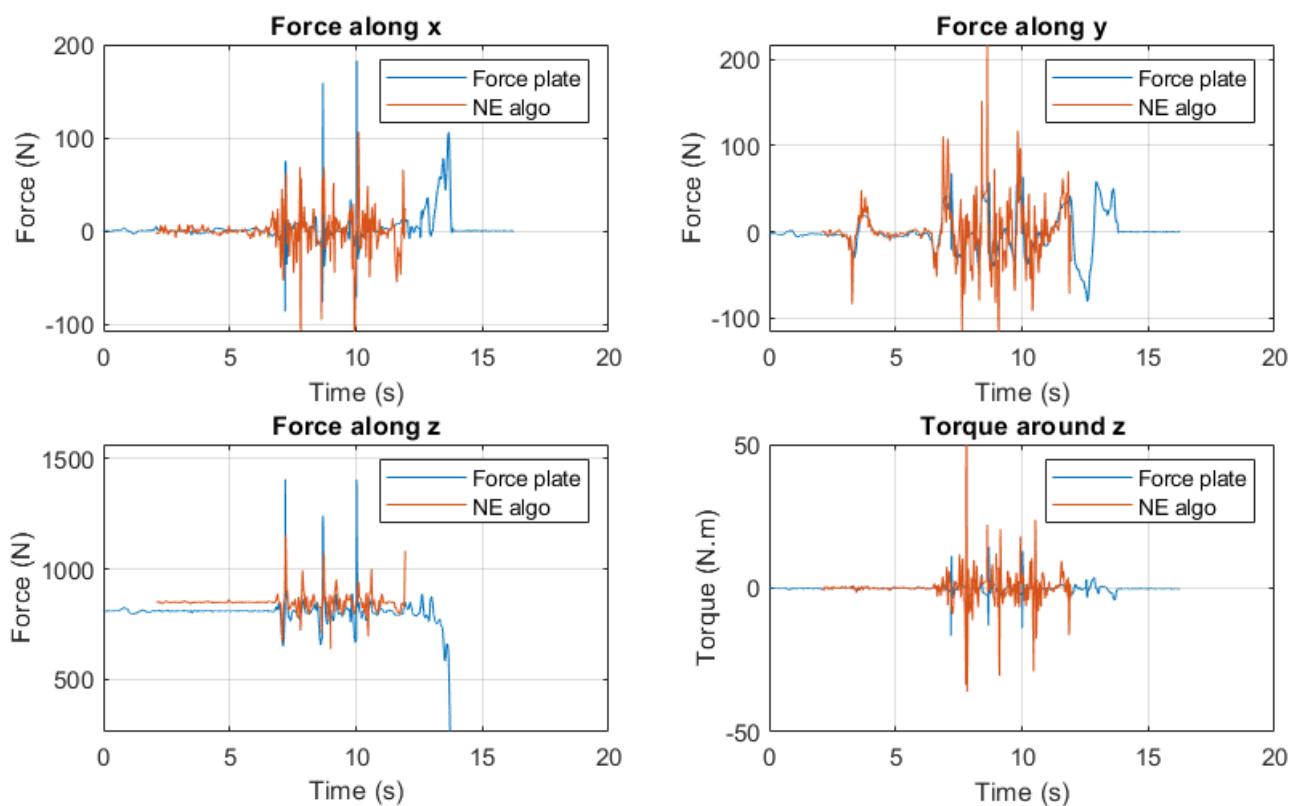


Figure 40: Fast arm

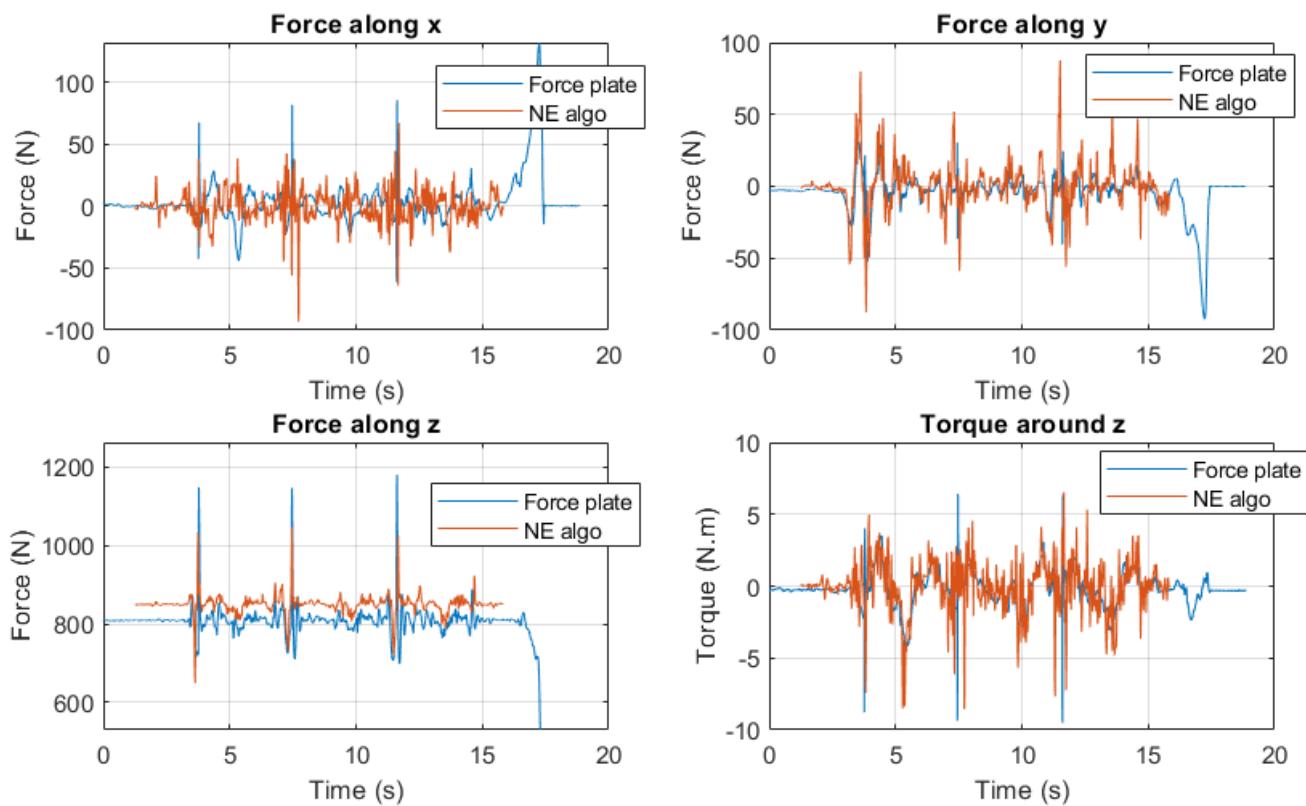


Figure 41: Slow kick

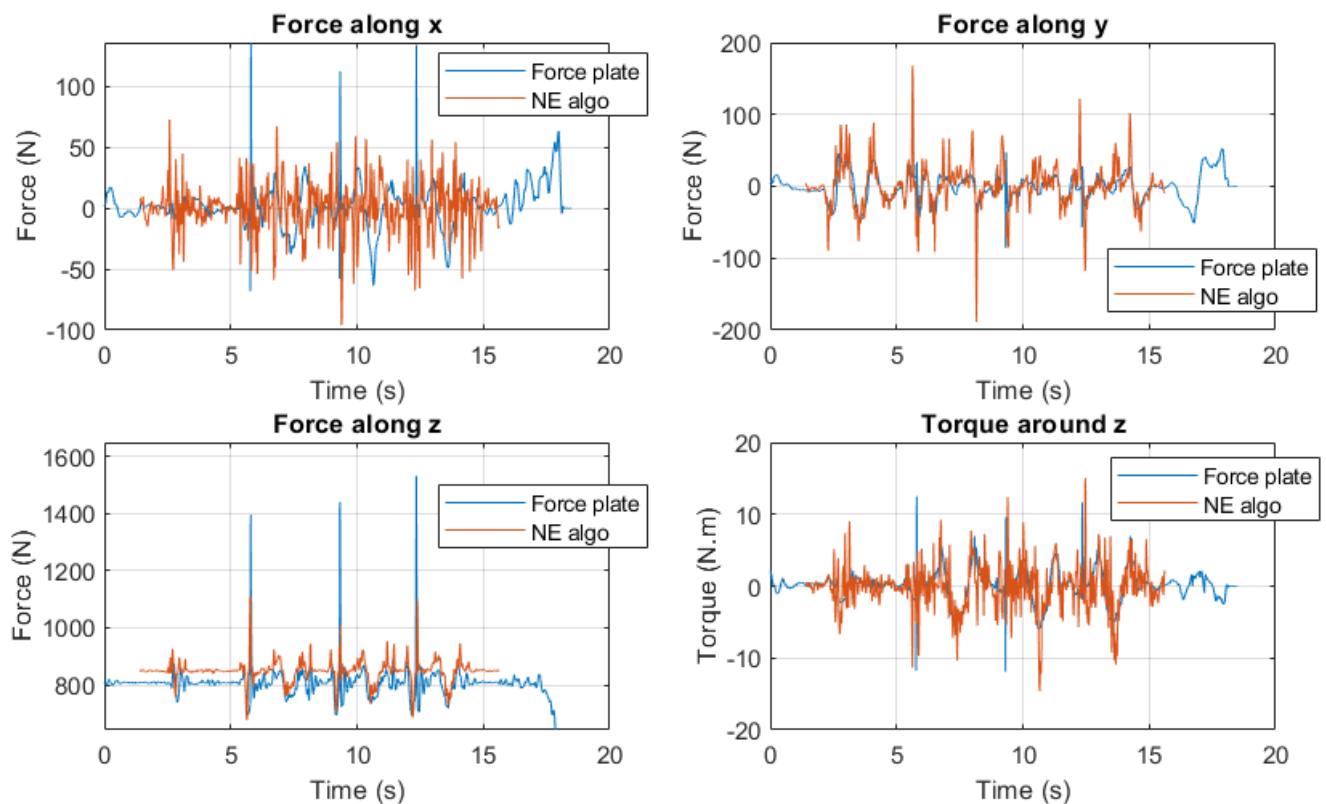


Figure 42: Medium kick

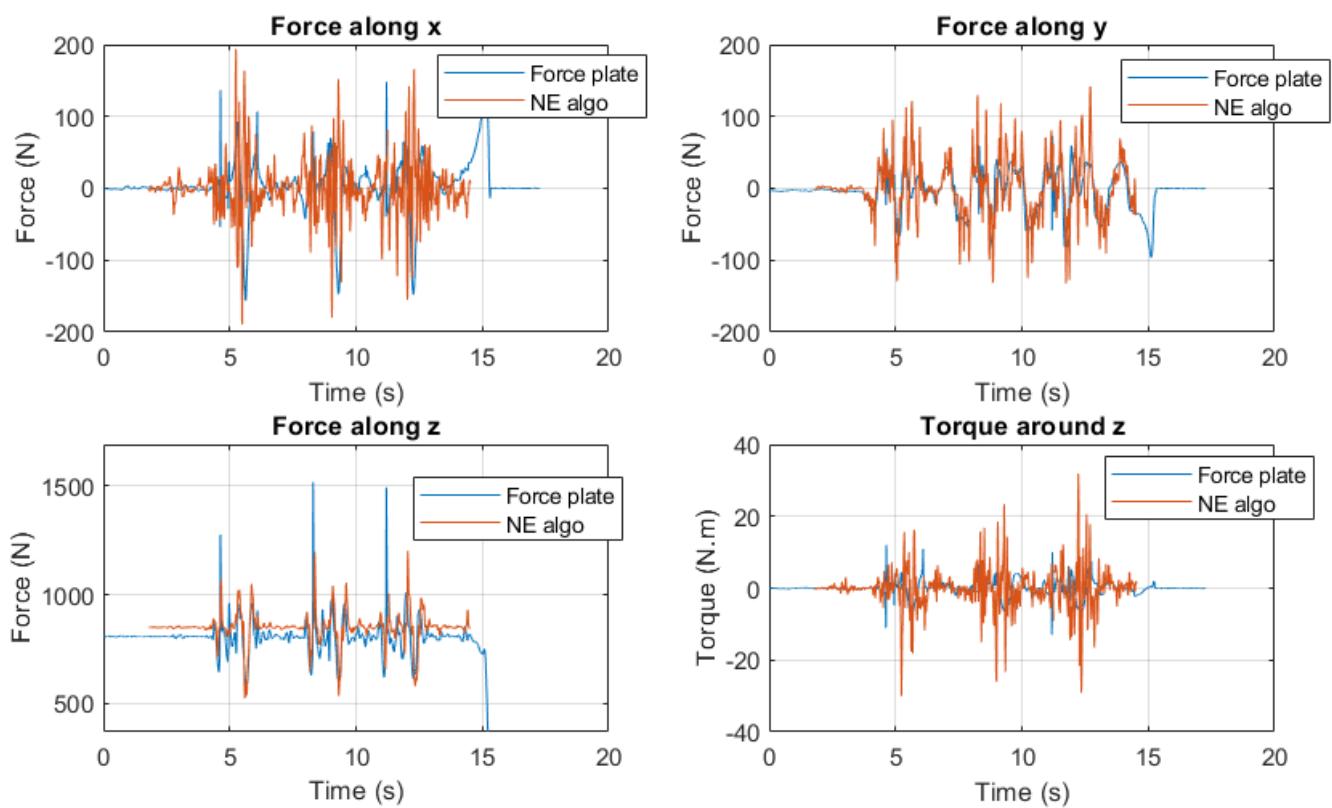


Figure 43: Fast kick

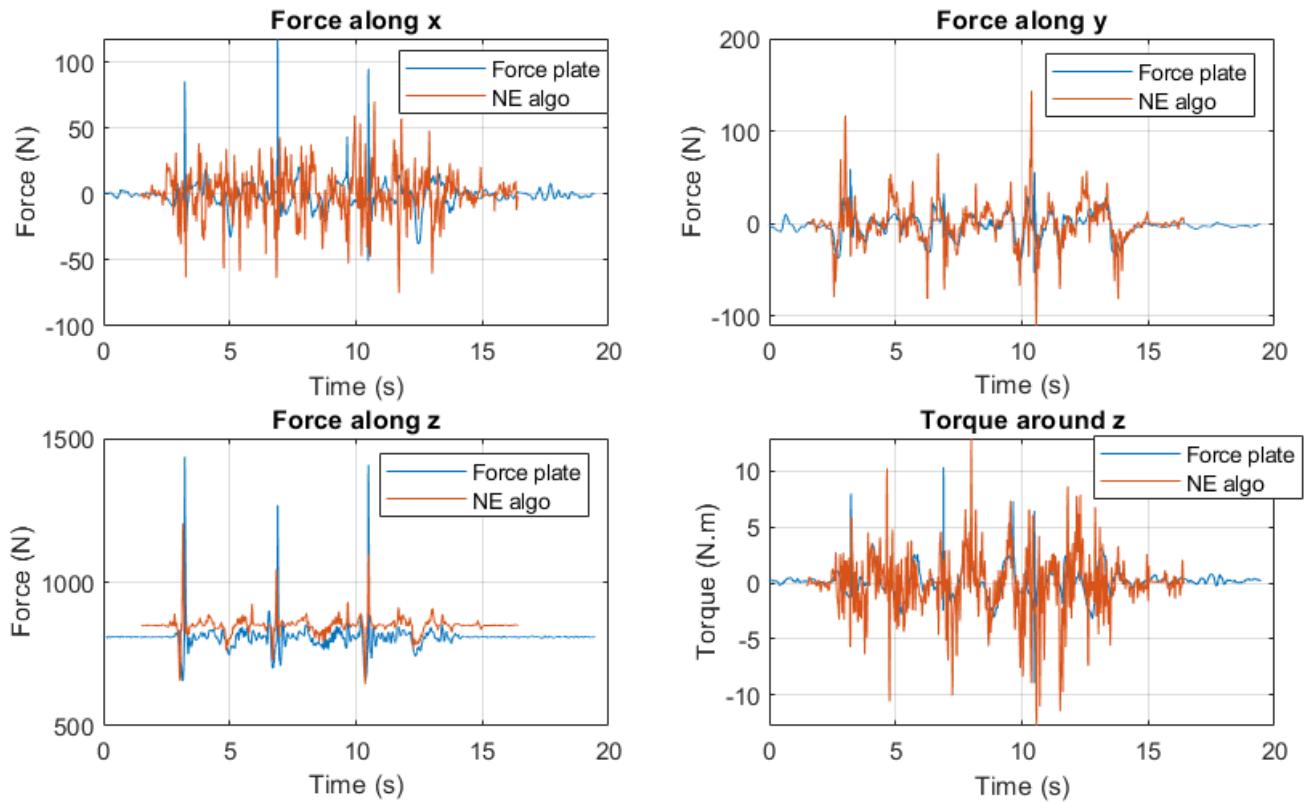


Figure 44: Slow kick arm

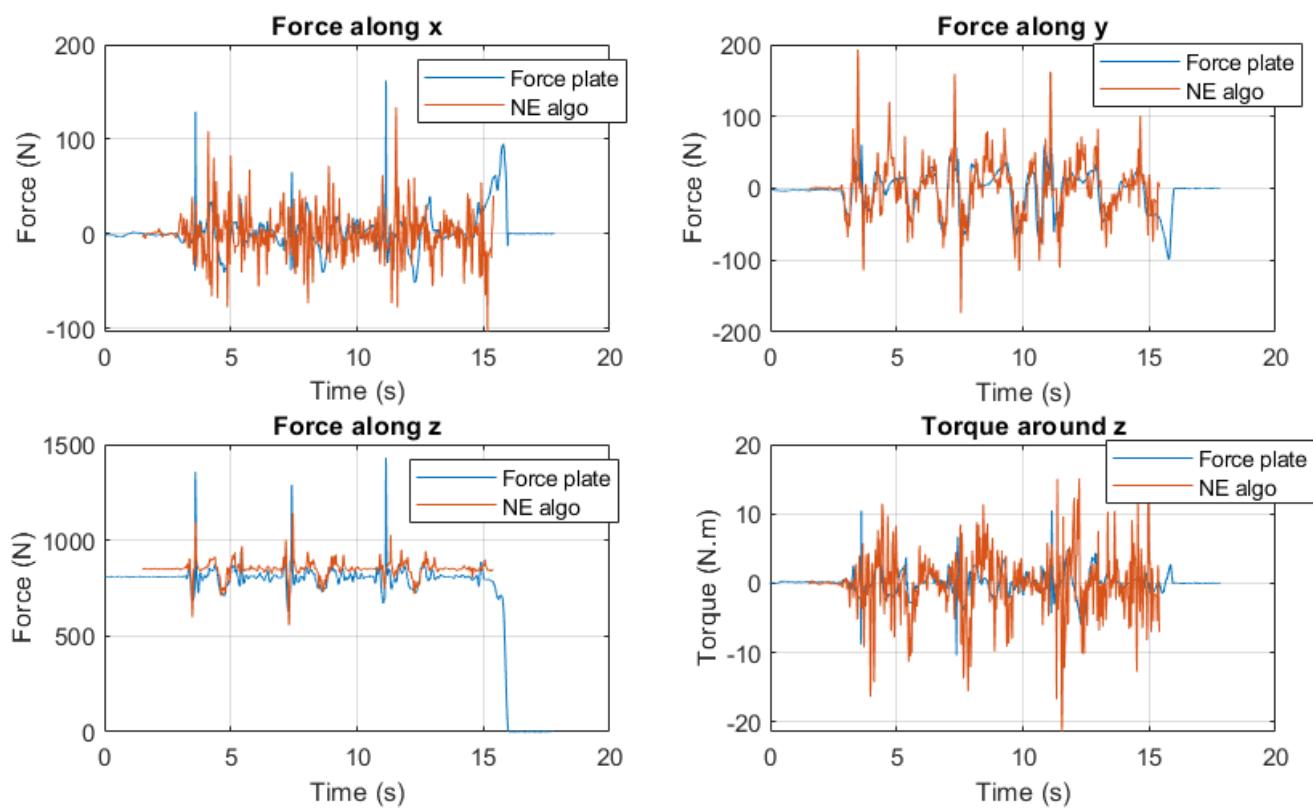


Figure 45: Medium kick arm

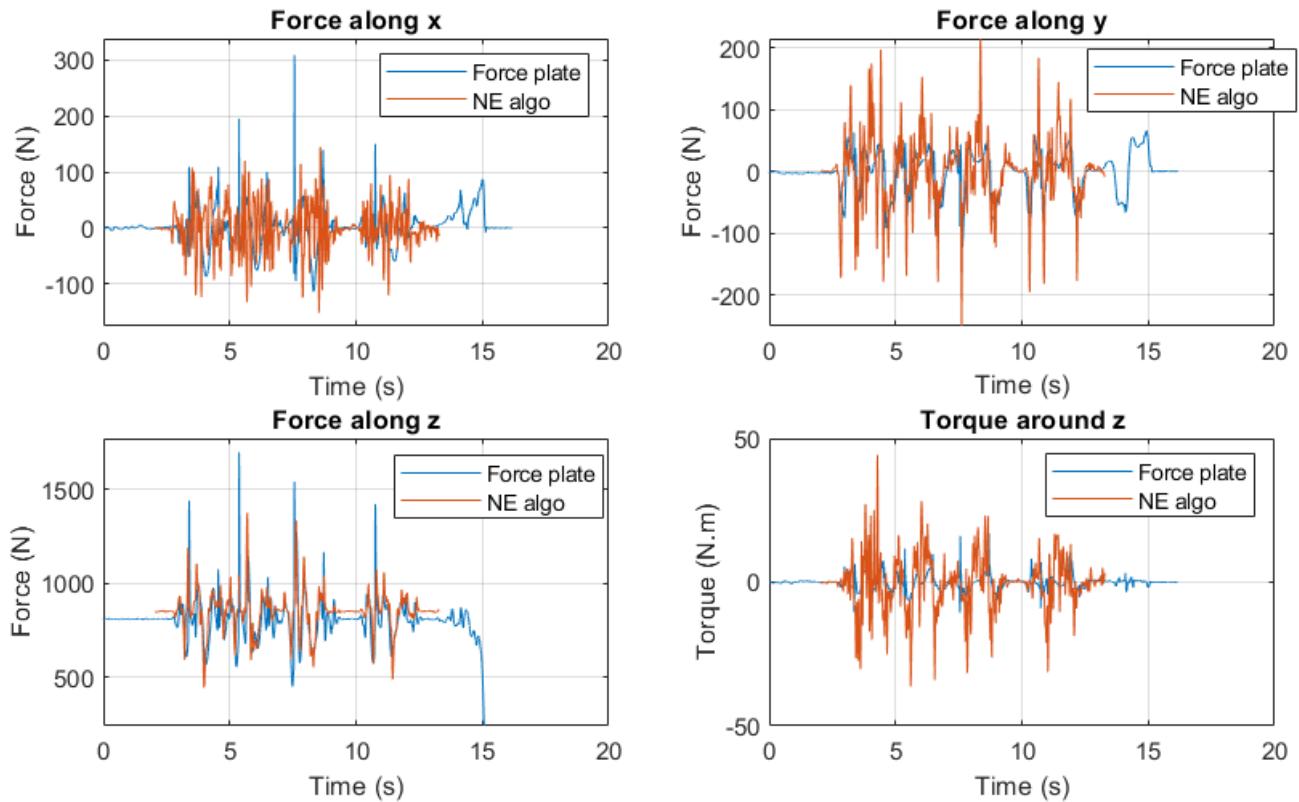


Figure 46: Fast kick arm

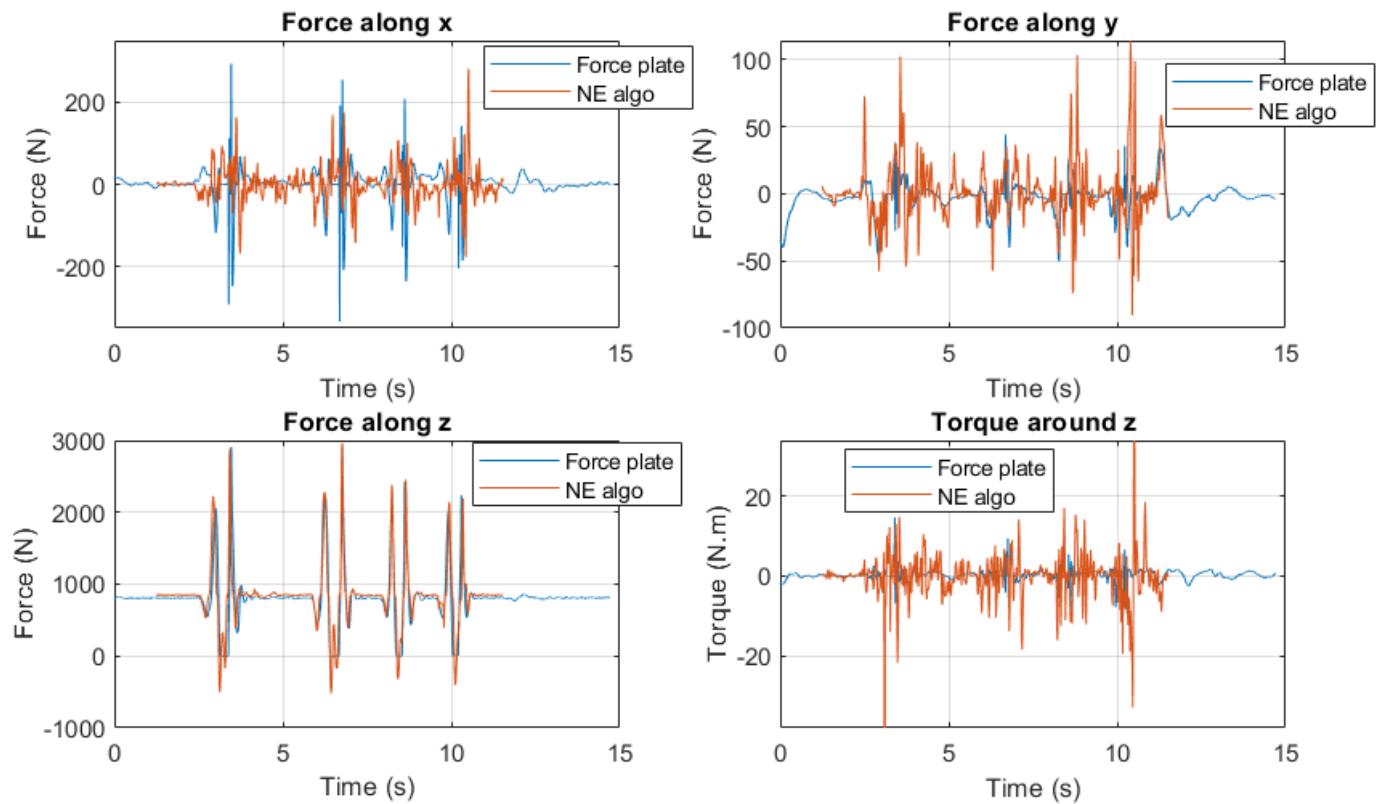


Figure 47: Quick jump

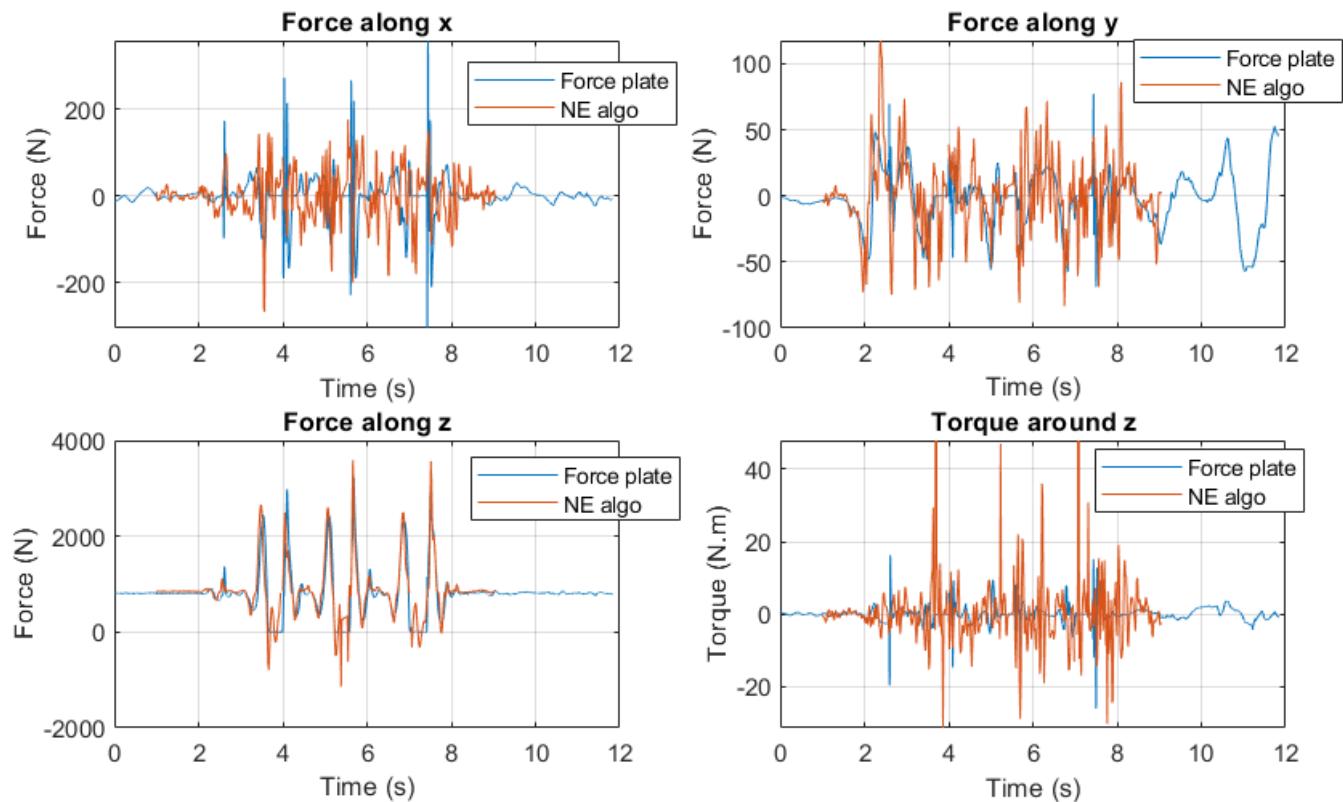


Figure 48: Medium jump